Поддержка: стоит ли платить? с. 44

LINUX FORMAT

Главное в мире Linux

Август 2007 № 8 (95)

LXF DVD



Fedora 7
и PCLinuxOS 2007

Плюс: более 70 отборных приложений

Управляй KDE из скриптов! с. 74

СПРАВОЧНИК ЖЕЛЕЗО—2007



Ничего не покупайте, пока не прочтете это! с. 32

GPLv3 уже с нами!

Брюс Пиренс, Эрик Реймонд делятся своим мнением с. 40

Потерялись в Linux?

Закажите экскурсию по файловой системе с. 48

Gentoo 2007.0

Наш вердикт: 9 из 10! А чем он может быть полезен вам? с. 14



“Люди устали от труда по поддержанию Windows”

Джим МакКиллан о перспективах LTSP с. 28

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882
Каталог «ПРЕССА РОССИИ» – подписной индекс 87974

К Вашим услугам...

В этом номере много говорится о GPL3, поэтому мы решили спросить у Команды LXF: какой еще пункт вы хотели бы видеть в ее тексте и почему?



Пол Хадсон
Положение, еще больше затуманивающее разницу между «free beer» и «free speech» может вылиться для меня в дармовую кружечку.



Грэм Моррисон
Запрет Ричарду Столлмэну использовать слова «свобода» и «личная» в одном предложении.



Эфрейн Эрнандес-Мендоса
Требование принимать душ каждый день, когда вы пользовались открытым ПО.



Ребекка Смолли
Пункт, согласно которому свободное ПО не только может, но и обязано распространяться. Плодитесь и размножайтесь!



Энди Ченнел
Положение вроде: «Я, нижеподписавшийся, настоящим заявляю тыры-пыры, на неограниченный срок. Ну, вы в курсе всякой юридической лабуды.»



Майк Сондерс
Нужен Санта. Вся эта юридическая скукотень читалась бы куда веселее, если б какой-нибудь толстопузый парень в красной шубе впихнул бы пару «хо-хо-хо» между строк.

Миссия журнала

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО



Прочтите перед использованием

В этом номере мы снова будем говорить о совместимости Linux с оборудованием – а именно, рассмотрим представителей различных классов периферии и решим, какие из них подходят для вашего Linux-компьютера, а от чего лучше держаться подальше. Я уже готов принимать письма от недовольных читателей – поскольку приведенный перечень будет неполон по определению, а простое перечисление всех аспектов работы лишь с наиболее распространенными устройствами займет не одну сотню страниц. Естественно, наш журнал далеко не пионер в области написания всевозможных справочников покупателя (откройте любое околокомпьютерное издание), но я бы хотел обратить внимание на одну деталь: во всех других случаях о совместимости с операционной системой говорят в последнюю очередь. Оно и понятно – сложно найти устройство, которое не заработало бы под Windows, пусть даже после некоторых не вполне ясных манипуляций. Радует, что с каждым месяцем это становится все более и более справедливым и в отношении Linux, однако, до «вставил – установил драйвер с диска – начал работать» еще далеко: идеология «все свое ношу с собой», присущая Linux, здесь оборачивается против нас. Так что же лучше: раздувать ядро, включая в него все новые и новые драйверы или же разработать удобный механизм их установки в систему, не завязанный на конкретику дистрибутива? Мы ждем ваших писем или сообщений на форуме! **LXF**

Валентин Синицын » Главный редактор info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru
› **Адрес редакции:** Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54.
› **Телефон редакции:** (812) 717-00-37. Дополнительная информация на стр.120

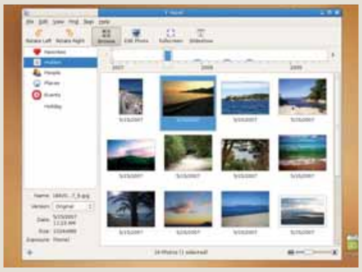
LINUX FORMAT

Август 2007 № 8 (95)

Содержание

Весь номер – прямо как на ладони: приятного чтения!

Учебники



› Поделитесь впечатлениями!



LXF DVD95



Fedora 7
Уже не Fedora Core, но по-прежнему добротный дистрибутив. Теперь – с быстрой сменой пользователя, горячим подключением устройств отображения и новым сетевым инструментарием.

PCLinuxOS 2007
Восходящая звезда на небосклоне Linux, PCLinuxOS работает в режиме LiveCD, так что вы можете посмотреть на новейшее ПО без необходимости что-либо устанавливать.

Журнал в PDF
Как влиться в открытый проект? Как настроить MythTV? Почему в мире Open Source так мало женщин? Ищите ответ здесь!

И многое другое...
Воспользуйтесь оболочкой диска, чтобы найти средства разработки, настольные приложения, аудио-утилиты, драйверы, игры...



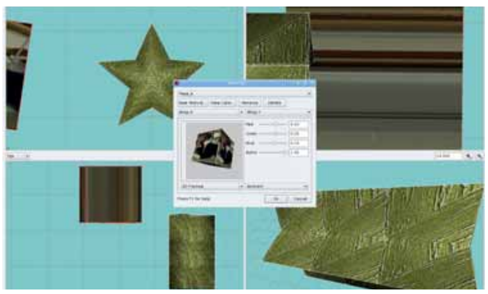
› Шикарная новая тема Fedora 7 как нельзя лучше подходит к такому классному релизу.

Что за штука... JavaFX?

Новое в web-разработке... с. 52



LXFHotPicks



› Пологая кривая обучения 3D-моделированию.



LXF ИНТЕРВЬЮ

«Те, кто пользуется LTSP, довольны им.»

Джим МакКиллан с. 28



Секретная операция с. 44

Лицом к лицу со службами поддержки Red Hat, Novell и Ubuntu.



Подпишись на Linux Format и сэкономь!

Содержание



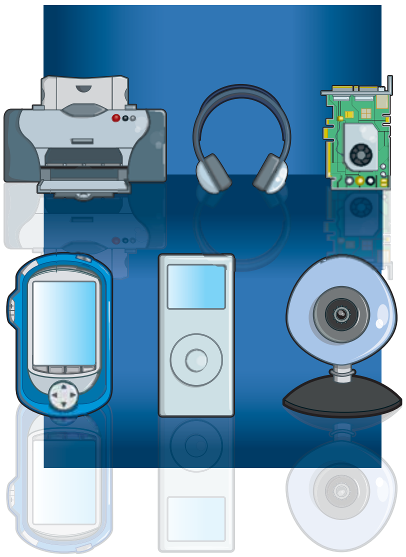
LXF DVD с двойной загрузкой!

Хотите узнать, что на нем? См. с. 104

Спецрепортаж

Справочник Железо—2007

Хит-парад дружественных к Linux устройств! с. 32



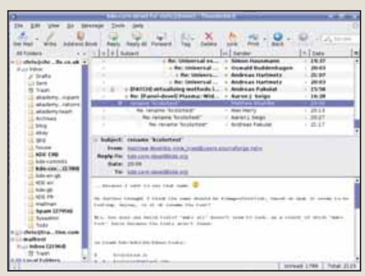
А также...

Обзоры



› Самые быстрые гонки под Linux – любые другие вам теперь покажутся замедленной съемкой...

Сравнение: почтовые клиенты



Постоянные рубрики

› Что-то новенькое на стр. 56!



LINUX FORMAT Новости

ГЛАВНЫЕ НОВОСТИ: » Встречайте GPLv3 » Российской ОС – быть? » Корпоративная Mandriva » Террористы выбирают Linux? » ODF и OOXML в Азии » Чаши с яблоками или яблоки в чашах? » Slackware 12.0

» Рубрику ведет
Илья Шпаньков



GPLv3 – новая эпоха

29 июня 2007 года завершилась разработка третьей редакции свободной лицензии *GPL*. Событие это, безусловно, станет одним из ключевых в истории свободного ПО, последствия которого пока трудно прогнозировать. За полтора года, в течение которых шла подготовка новой редакции, было сломано немало копий в спорах о содержании этого основного документа для всего сообщества пользователей и разработчиков *Free Software*. Но, как бы то ни было, совместными усилиями завершена работа, и в ближайшем будущем станет понятно, насколько точно предугадали создатели *GPLv3* дальнейшее развитие ИТ-индустрии и сможет ли новая редакция выполнить свою роль – защитить свободное программное обеспечение от чрезмерной коммерциализации и вытекающих из нее ограничений. Впрочем, многие крупные поставщики Linux и компании, активно использующие и продвигающие свободное ПО, уже высказали свое одобрение по поводу нового документа. Одними из первых озвучили свою поддержку IBM, Red Hat и Novell, а также *MySQL AB*. В частности, представители «голубого гиганта» оценили новую редакцию *GPL*, как «абсолютно коммерчески жизнеспособную», в связи с чем компания IBM будет активно популяризировать новую версию среди своих клиентов. Менее масштабные компании и свободные проекты также высказали свое одобрение новой свободной лицензии, но сделали это без громких слов – уже в первые дни после релиза *GPLv3* десятки проектов (в том числе, *GNU Core Utilities*, *GnuPG*, *PeaZip*, *Tar*, *Texinfo*, *Sed*, *WeeChat*) перевели свои программные продукты на использование новой редакции. Не отстали и разработчики Samba – все версии, начиная с 3.2 будут лицензированы именно по *GPLv3*. Что ж, очередной эволюционный виток завершился и начался новый, который, надеемся, позволит свободному ПО достичь новых высот.

www.gplv3.fsf.org

МИНИСТЕРСТВО ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СВЯЗИ РОССИЙСКОЙ ФЕДЕРАЦИИ реабилитируется

Неутомимый депутат Госдумы РФ Виктор Алкснис, ставший в последнее время своеобразным представителем и защитником интересов свободного ПО в парламентских коридорах, в начале июля опубликовал в сети официальный ответ Министерства информационных технологий и связи на думский запрос относительно позиции данного ведомства по отношению к созданию отечественной операционной системы на базе Linux. Ко всеобщему удивлению, весьма подробный ответ, развернувшийся на 15 страниц, показал, что Мининформсвязи не только поддерживает идею создания российской операционной системы, но и активно занимается реализацией этого проекта. В документе подробно рассматриваются все возможные кандидатуры на отечественную «ось», начиная от полусекретных разработок для ФСБ и Вооруженных сил РФ и заканчивая молодыми публичными проектами, пока только набирающими определенный вес в сообществе. Примечательно, что практически все дистрибутивы в своей основе содержат код операционной системы Red Hat: в список клонов входят ведомственные МСВС (Мобильная Система Вооруженных Сил), Атликс (используется в ФСБ) и Yanux, а также «гражданские» ASPLinux и другие (всего шесть вариантов). И лишь один дистрибутив AltLinux оказался достаточно «возмужавшим», чтобы содержать собственный репозиторий Сизифус (получивший недавно, кстати, заслуженное признание от международного сообщества). Впрочем, несмотря на довольно внушительный список, чиновники пока не видят реального кандидата, готового к широкомасштабному внедрению на всей территории России. Помочь в скорейшем появлении подобной разработки и призвана программа Мининформсвязи, которая помимо вопросов создания операционной системы подразумевает и продвижение открытого международного стандарта формата документов ODF в официальном документообороте.

www.v-alksnis2.livejournal.com/42625.html

Mandriva Corporate: скромные, но все-таки успехи

Постоянная нацеленность компании Mandrakesoft на рынок настольных систем приводила к тому, что в новостях, касающихся корпоративного сектора, ее имя практически не упоминалось. Но, похоже, ситуация постепенно меняется: вслед за выпущенной в прошлом году операционной системой Mandriva Corporate Server 4.0 разработчики одного из самых популярных дистрибутивов к лету текущего года подготовили и версию Mandriva Corporate Desktop 4.0, предназначенную для установки на рабочие машины крупных коммерческих предприятий и организаций. Одним из основных преимуществ программных продуктов данной линейки могут служить перешедшие по наследству от персональных версий операционных систем Mandriva хорошо развитые графические средства настройки и управления, а также удобный стильный интерфейс. Данный фактор может помочь корпоративным версиям дистрибутивов Mandriva в продвижении на рынке – программная продукция признанного лидера в данном секторе компании Red Hat особыми красотами не блещет, предлагая пользователям надежный, но немного аскетичный в плане внешнего оформления инструментарий.

Впрочем, успешность программного продукта определяет не красота интерфейса, а востребованность на рынке. Что ж, в этом плане у Mandriva тоже есть определенные достижения, хотя и более скромные, чем у конкурентов в лице Red Hat и Novell. В частности, Министерство сельского хозяйства и рыболовства Франции выбрало серверный вариант Mandriva Corporate Server 4.0 для осуществления миграции на свободное ПО. Программа рассчитана на два года, в течение которых около 400 серверов министерства будут переведены с Windows NT на данную операционную систему. Одновременно Mandriva берет на себя всю работу по технической поддержке своего ПО и обучению персонала. В общей сложности планируется, что более двухсот работников министерства получат навыки работы и управления данным решением в авторизованных центрах обучения Mandriva. Стоит также отметить, что планы Mandriva по продвижению своих корпоративных версий не ограничиваются пределами родной Франции: вслед за российским филиалом в июле текущего года открылось еще одно представительство компании – на этот раз в Великобритании. Данные шаги говорят о том, что в планах Mandriva вопросам завоевания корпоративного сектора рынка уделяется немаловажное значение. И это можно только приветствовать: корпоративный потребитель так же, как и обычные домашние пользователи, должен иметь свободу выбора.

Особенности национальной радиофикации



Маниакальная склонность американских чиновников к привычке раскладывать по правовым полочкам каждую мелочь уже давно вызывает улыбку у большинства европейских жителей. И очередной пример, продемонстрированный Федеральной комиссией связи США (FCC), тому подтверждение. Данная солидная организация подготовила новые инструкции, запрещающие разработчикам устройств, использующих радиочастоты, раскрывать исходный код соответствующего программного обеспечения. Основная причина – желание чиновников ограничить возможности потенциальных террористов по перенастройке устройств на использование запрещенных или ведомственных радиочастот. Нельзя сказать, что подобная идея является чем-то новым: прошивки для большинства беспроводных сетевых карт и так поставляются только в двоичном виде, что как раз объясняется желанием затруднить изменение частоты, на которой работает устройство. Однако, последствия подобной перестраховки Федеральной комиссии могут довольно неприятным образом сказаться на компаниях, использующих в своих изделиях открытое ПО, а число подобных производителей сегодня достаточно велико: дело в том, что практически все мобильные телефоны и адаптеры беспроводной связи работают именно с радиочастотами, что автоматически подразумевает следование инструкциям чиновников. При этом представители федеральной комиссии совершенно не учитывают тот факт, что обычный пользователь сотового телефона или оборудованного WiFi ноутбука и так не страдает большим желанием править исходные тексты встроенного ПО, а для преступников данное ограничение не имеет решающего значения. Впрочем, если же основной целью чиновников было создать дополнительные трудности на пути внедрения открытого ПО, то тогда данные инструкции выглядят вполне логично.

«Форматирование» по-азиатски

Очередные новости на тему соперничества двух открытых форматов – ODF и OOXML – приходят в последнее время из азиатского региона. OpenDocument Format Alliance объявил о том, что рекомендации по внедрению открытого международного стандарта ODF приняты правительством Японии и доступны для ознакомления на web-сайте Министерства экономики, торговли и промышленности. Власти данной страны уже давно проявляли интерес к открытым стандартам и свободному ПО, что подтверждают многочисленные проекты, поддерживаемые или инициированные правительством Японии. Теперь же переход на открытые международные стандарты стал общенациональной программой, в процессе реализации которой будут полностью ликвидированы потенциальные нестыковки во взаимопонимании как различных ведомств, так и граждан страны со своими властями.

Практически одновременно и компания Microsoft, продвигающая свой открытый формат документов OOXML, объявила об определенных успехах в азиатском регионе. В частности, редмондский гигант заключил соглашение с крупнейшим разработчиком Linux-систем японской компанией Turbolinux, в рамках которого азиатский партнер примет активное участие в разработке ODF-OOXML конвертера, а также будет способствовать активному продвижению формата OOXML на японском и китайском рынке ПО. Задача, в общем, понятна: чем большее распространение получит данный формат, тем проще будет уговаривать японских и китайских бизнесменов на покупку пакета офисных приложений Microsoft Office, который, собственно, и приносит редмондскому гиганту основную прибыль.

Встречаем Slackware 12.0

Бессменный руководитель проекта Slackware Патрик Фолькердинг [Patrick J. Volkerding] в июле порадовал своих поклонников долгожданной новой версией дистрибутива Slackware 12.0. В числе наиболее явных изменений можно отметить переход на использование ядра Linux 2.6.21, внедрение системы HAL (Hardware Abstraction Layer), включение в состав дистрибутива набора свободных компиляторов *GCC 4.1.2*. В качестве графических оболочек предлагается KDE 3.5.7 и *XFCE 4.4.1*. Следует отметить, что данный релиз содержит большое число наиболее свежего программного обеспечения, в числе которого библиотеки *glibc 2.5*, графический сервер *X.Org 7.2.0*, *Apache 2.2.4*, *PHP 5.2.3*, *Perl 5.8.8*, *Python 2.5.1*, *Ruby 1.8.6*, *Subversion 1.4.4*, *SeaMonkey 1.1.2*, *Firefox 2.0.0.4*, *Thunderbird 2.0.0.4*, *Pidgin 2.0.2* (бывший *Gaim*), *The GIMP 2.2.15* и многое другое.

› Ищите *Slackware 12.0* на прилагаемом к журналу диске!
www.slackware.com/announce/12.0.php

Apple, повелитель принтеров



Всередине июля автор популярной в UNIX системы печати Common UNIX Printing System (CUPS) и по совместительству глава компании Easy Software Products Майкл Свит [Michael R. Sweet] обнародовал подробности своей сделки с компанией Apple Inc. Суть в том, что в феврале текущего года производитель Mac'ов и iPod'ов приобрел у Свита все права на исходный код CUPS, тем самым взяв под свое покровительство самую популярную подсистему печати, входящее по умолчанию практически во все дистрибутивы GNU/Linux. Названия "CUPS", "Common UNIX Printing System" и логотип CUPS также стали зарегистрированными товарными знаками Apple Inc. Сам Свит при этом принят в штат компании в качестве ведущего разработчика CUPS. Сумма сделки, естественно, не разглашается, но от объяснений по поводу дальнейшей судьбы системы печати, распространяемой под лицензиями *GPL*/*LGPL*, автор проекта решил не отказываться, и постарался максимально успокоить заволновавшееся сообщество. Ко всеобщему удовлетворению, код CUPS по-прежнему будет распространяться под вышеназванными лицензиями. Это позволит разработчикам дистрибутивов и дальше включать данный пакет в состав своих программных продуктов. Что же касается торговых марок, то их использование также остается свободным, за вычетом случаев их включения в производные программные продукты – в этом случае будет достаточно получить письменное разрешение от Apple Inc. Теперь о выгоде. С автором CUPS все понятно – он получил вполне реальные денежные средства, которые, надеемся, помогут Свиту с удвоенной энергией совершенствовать свое детище. Что же касается сообщества Free Software, то его представители надеются получить в лице Apple Inc. нового влиятельного покровителя свободного ПО, в числе которых уже значатся такие имена, как IBM, Hewlett-Packard и Novell. Также не исключено, что хорошая техническая и экспериментальная база «яблочной» компании позволит значительно улучшить систему печати в операционных системах GNU/Linux.
www.cups.org/articles.php?L475

Горячие новости из Якутии

В Информационном центре при Президенте Республики Саха прошли мастер-классы, посвященные свободному ПО, на которых присутствовали представители бизнеса, власти и образовательных структур Якутии. В качестве «наглядного пособия» использовалась операционная система Mandriva Discovery 2007 (одним из организаторов мероприятия выступил российский филиал Mandriva.ru), которую слушатели могли свободно устанавливать на свои компьютеры. Примечательно, что результаты данного мероприятия оказались на удивление оперативными: в учебном классе Информационного центра при Президенте Республики Саха на всех компьютерах был установлен дистрибутив Mandriva Linux 2007 Spring в качестве второй операционной системы для постоянного использования в дальнейшем. Одновременно была достигнута договоренность об открытии центров компетенции в Информационном центре при Президенте Республики Саха и в Якутском Государственном Университете. Также организаторы мастер-класса успели принять участие в телемосте с министром образования Якутии, где среди прочих вопросов прошло обсуждение проблем лицензирования ПО в школах республики. Пожалуй, на сегодняшний день это один из наиболее плодотворных итогов обычных, казалось бы, разговоров на тему свободного ПО.



Фото: http://internetmarafon.ru/

www.linuxcenter.ru/lib/press/mandriva_saha.phtml

Новости короткой строкой

» Начавшийся было конфликт между разработчиками *Wine* и компанией SWSoft по поводу нарушения GPL был оперативно улажен – проекту были переданы незаконно закрытые модификации.

» Компания Trolltech решила привлечь к работе над своими проектами разработчиков свободного ПО, предложив вместе с программными средствами и телефон Qtopia Greenphone.

» 57% серверов кандидатов в президенты США работают под управлением свободного ПО.

» Компания Id Software пообещала в ближайшее время выпустить Linux-версию клиента для игры *Enemy Territory: Quake Wars*.

» Число американских разработчиков открытого ПО за прошедший год возросло с 8,8% до 11,8%, в то же время программистов для Windows уменьшилось с 74% до 64,8%.

» Американская ассоциация кинокомпаний (MPAA) использовала подставные web-сайты и «троянов» для выявления пользователей нелицензионных медиа-данных.

» Правительство Голландии издало «Декрет о качестве голландских правительственных web-сайтов» в целях максимального соответствия последних стандартам W3C.

» По данным Australian Government Information Management Office (AGIMO), более 68% австралийских чиновников уже используют в работе свободное ПО.

LXF Репортаж

Неделя открытых технологий в Новосибирске

Пытаясь попасть на это мероприятие, **Евгений Балдин** простоял час в автомобильной пробке Не напрасно ли? Как выясняется, нет.

Работайте, работайте – а понимание придет потом.
Жан Лерон Д'Аламбер

Когда я ехал из Академгородка, расположенного на окраине города, в центр на мастер-класс в Новосибирском государственном техническом университете (НГТУ http://info.nstu.ru/), цивилизация дала о себе знать посредством часовой пробки на дороге. Я был не единственным, кто задержался в пути. К первому слушанию основной докладчик – директор Линуксцентра Павел Фролов также не успел. Всему виной туман. Поэтому первое собрание вышло немного скомканным.

Принимающая сторона, группа LinuxNSK (http://www.linuxnsk.ru) и партнер Линуксцентра фирма «Связь Комплект» в лице Алексея Рожанкова, узнавшего о существовании GNU/Linux всего месяц назад, постаралась выправить положение. Первый мастер-класс был направлен на представителей бизнеса, то есть фактически на тех, кто читает ЛОР (http://www.linux.org.ru/) и работает администраторами. Возрастной и гендерный состав весьма полно соответствовал ЛОР-срезу.

Презентация вышла весьма стандартной, оживило ее разве что предложение Антона Банщикова (инициатора «Неделя технологий GNU/Linux и Open-Source в Новосибирске») продемонстрировать работоспособность 1С под GNU/Linux. Лично я с пользой провел время, договорившись о трех «историях успеха», поприставав с разными вопросами к декану Факультета прикладной математики и информатики Борису Юрьевичу Лемешко, давшему терминальное помещение под проведение мастер-классов, и поболтав с ребятами из LinuxNSK. Мне понравилась эта группа с ее желанием что-то реально сделать. Пока на их счету перевод одной школы на открытое программное обеспечение, и у меня сложилось мнение, что это только начало. После мастер-классов силами LinuxNSK планируется проведение бесплатных ознакомительных курсов. Думается, подобное хобби – весьма достойное занятие для молодых людей, причем оно вполне может оказаться в будущем бесценным в смысле приобретенного опыта. Лучший способ разобраться с какой-то областью знаний – это попытаться передать эти знания другим людям. Только так можно досконально разобраться со всеми нюансами.

Примерно через час после окончания лекции появился основной докладчик Павел Фролов, сразу приступив к монтажу наглядной агитации, которая ко всем прочим достоинствам очень органично прикрыла часть окна. Это значительно улучшило качество картинки от проектора на экране. Погода была замечательной, и солнце светило на полную катушку – где там был туман, я так и не понял.

На следующий день я уже слушал доклад таким, каким он должен был быть. Аудитория поменялась – сидели серьезные дяди, которые хотели серьезных решений, так как в ВУЗах (второй день был посвящен образованию) с программным обеспечением в последнее время возникли не менее серьезные проблемы.

Вкратце выступление Павла Фролова можно охарактеризовать как рекламу решений Mandriva Linux (подробности о предложениях Линуксцентра можно узнать на его сайте http://www.linuxcenter.ru/). Собственно говоря, эта цель и не скрывалась, а трехмерный куб и «желеобразные окна» от Compiz на презентационном компьютере ей весьма способствовали. Со слов докладчика, дистрибутив Mandriva был выбран не случайно, а с учетом консультаций с потенциальными клиентами. Основными требованиями были «дружелюбность для пользователя» и качественная поддержка оборудования. Среди потенциальных клиентов рассматривались и федеральные структуры, то есть от геополитического фактора уйти не удалось. Российские дистрибутивы также учитывались при выборе, но отпали в основном по причине отсутствия тесных связей с производителями оборудования.

Системной поддержки GNU/Linux на территории России практически нет, и активная поддержка Mandriva со стороны Линуксцентра вполне может оказаться решением этой проблемы. Да, можно купить коробку, например, RHEL и получать поддержку по электронной почте или по телефону, но электронная почта будет на английском языке, а звонок будет за







границу, и языком общения будет отнюдь не русский. Да, действительно всесторонне развитый «Homo Adminus» обязан знать английский язык в совершенстве, но в любом случае задавать вопросы на родном языке дорогого стоит, так как позволяет думать собственно над вопросом, а не над его произнесением. Кроме того, местная поддержка волей-неволей будет вынуждена изучить местные же особенности и потребности, что облегчает решение локальных проблем.

Конкретно школам и ВУЗам посулили возможность купить академическую лицензию примерно за 30 тыс. руб в год, которая дает полную поддержку от Линуксцентра и право на использование дистрибутивов внутри учебного заведения, а также дома всем его учащимся и преподавателям. Наличие возможности покупки официальной лицензии на программное обеспечение, то есть бумаги с синей печатью, должно оградить от потенциальных проверок со стороны компетентных органов. Вопрос о проверках в учебных заведениях вызвал неподдельный интерес со стороны слушателей. Тема на сегодня очень актуальна, и полное решение проблемы пока отсутствует. Павел Фролов сообщил о случаях, когда под давлением администрации с компьютеров, поставленным по федеральной программе информатизации, удалялось просто все программное обеспечение, что, естественно, приводило компьютерные классы в негодность. Поэтому в использовании открытого программного обеспечения многие из присутствующих надеялись увидеть вероятный выход. Вероятный, потому что далеко не всем этот выход нравится своей непривычностью. Жизнь учителя и так непроста, и любое дополнительное усложнение ему вовсе не в радость.

Некоторым из присутствующих очень хотелось рассказать о своих проблемах, но указать на корень всех зол не очень получилось. Было предложено сообщать обо всех имеющихся препятствиях на пути использования свободного программного обеспечения местного, областного и уж тем более федерального уровня по электронной почте лично Павлу frolov@mezon.ru. Со стороны докладчика ощущалось очевидное недопонимание базовых принципов, по которым строится организация школьного образования, и обратная связь для ликвидации этого недопонимания, по его уверению, будет использоваться с энтузиазмом. Так же выяснилось, что Академическая программа не распространяется на Академические центры, но было обещано, что в случае возникновения конкретных вопросов данная проблема будет обязательно рассмотрена. У меня как сотрудника РАН такие вопросы определенно возникнут – буду следить за развитием событий.



Третий и последний день мастер-класса не был ориентирован на какую-то определенную аудиторию, поэтому собрались просто желающие. Некоторые из них пришли во второй раз. После стандартной оды дистрибутиву Mandriva с клятвенными заверениями, что этот дистрибутив в Россию пришел всерьез и надолго, наступило время конкретных вопросов и ответов, часть из которых с моей точки зрения была довольно интересной. Любопытно было узнать, что Mandriva Flash (полноценный дистрибутив GNU/Linux, позволяющий более-менее комфортно работать «с» и сохранять свои файлы «на» USB-брелок) – это целиком российская разработка, сделанная в Санкт-Петербургском центре Mandriva, финансируемом Линуксцентром. Актуальной выглядит инициатива Линуксцентра по законодательному признанию GNU и BSD лицензий в России. Разговор касался не только GNU/Linux – было уделено внимание дистанционным технологиям (http://www.intuit.ru/) и детским обучающим программам (http://www.tirnet.ru/). Не знаю как другим, а мне понравилась Flash-игра «построить пингвиненку домик» – не так уж это и легко.

Значительная часть времени пришлась на рассказ про журнал Linux Format и тем сложностям, с которыми столкнулся Линуксцентр, когда попытался издать качественный журнал о GNU/Linux в России. Был призыв выписывать журнал, а не покупать в магазинах – в этом случае одна единица журнала и для читателя выходит дешевле, и самим издателям достается больше. К концу года пообещали организовать собственную вменяемую подписку на электронную версию журнала.

Интересно отметить, что почти никто из пришедших не спешил попробовать установленный на терминалах дистрибутив Mandriva. То есть, пришли люди, которые уже знали, что такое GNU/Linux, и им не было необходимости проверять слова докладчика. Гораздо больший интерес представляли сами эти люди. Такое ощущение, что системным администраторам, которые составляли значительную часть от участников, достаточно тяжело организовываться в группы, но все же интерес к собеседникам, как правило, высказывался совершенно неподдельный.

Я не пожалел, что сходил на это мероприятие. Конечно, многое можно было сделать лучше, но все мы крепки «задним умом». В любом случае было приятно увидеть представителя организации, реально занимающегося инновационной деятельностью в классическом понимании этого слова. Ну и «междусобойчики», которые традиционно возникали после каждого из мастер-классов, были достаточно наполнены, в том числе и в эмоциональном плане.

Побольше бы таких собраний, хороших и разных! LXF

Фотографии предоставлены Александром Рожанковым

Что будет: события сентября

VI Всероссийская конференция «Обеспечение информационной безопасности. Региональные аспекты»
Где: г. Сочи **Когда**: 11–15 сентября 2007 года

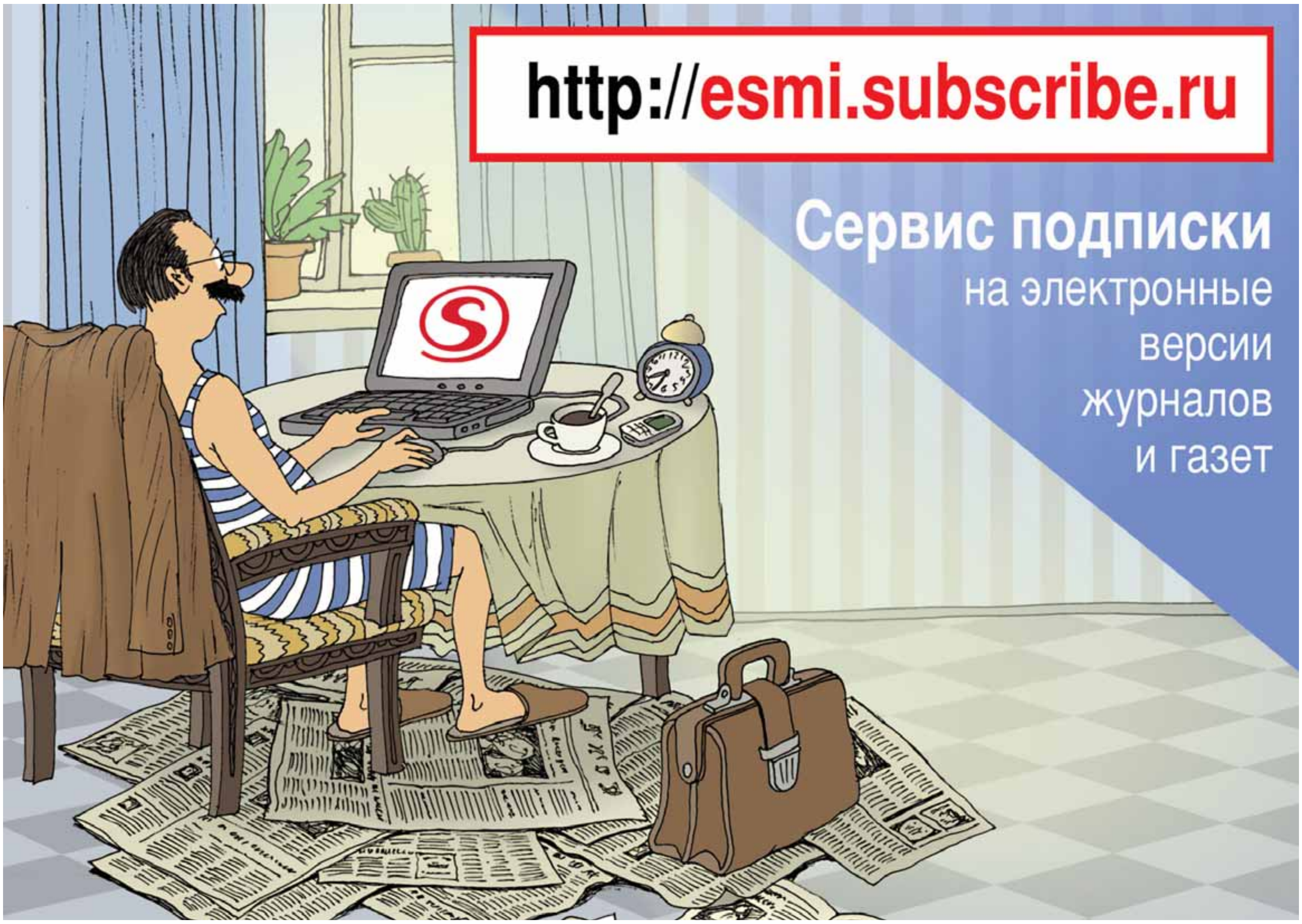
В конференции традиционно примут участие представители федеральных органов государственной власти, контролирующих структур, среди которых аппарат Совета безопасности РФ, Государственная Дума ФС РФ, Совет Федерации, аппараты полномочных представителей Президента РФ в федеральных округах, Мининформсвязи РФ, МВД России, Минпромэнерго, ФСТЭК, ФСБ, Росинформтехнологии, Ростехрегулирование, Роспром. 14 сентября впервые в рамках конференции будет представлена рабочая секция «Решения в области безопасности на базе Linux и Open Source». На секции выступят представители ведущих поставщиков в области безопасности на платформе Linux и открытом ПО.

http://www.infosystem.ru/longconf.php?fid=1178526429948711

Open Source Day 2
Где: г. Караганда, Казахстан **Когда**: 20 сентября 2007 года

На Open Source Day 2 планируется заслушать выступления казахстанских, российских и международных экспертов в области программного обеспечения с открытым кодом, представителей коммерческих компаний и государственных органов. Будет проведен практический семинар по использованию операционной системы Linux. Среди основных участников конференции: аппарат акима Карагандинской области, районные акимы, городские и районные отделы образования, компания ALTLinux, международные эксперты из Швейцарии, Литвы, Эстонии, России, представительство ЮНЕСКО, общественные организации, представительства иностранных IT компаний, учителя средних школ, студенты карагандинских ВУЗов, СМИ.

http://www.os.kz/index.php?p=553


http://esmi.subscribe.ru
Сервис подписки
на электронные
версии
журналов
и газет



Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



Алексей Федорчук
Свою первую (и последнюю) программу написал еще на Алголе

Не спрашивай, что ты сделал для Linux'а...

... ибо настало время спросить, что Linux делает для тебя. И это будет самая малость: он позволяет работать, не вникая в тонкости организации системы и ее настройки. Канули в Лету времена, когда первыми действиями были пересборка ядра, прикручивание кириллицы, прописывание параметров конфигурации Иксов, сочинение скриптов для выхода в Интернет, не говоря уже о сборке недостающих пакетов из исходных текстов. Нет, конечно, вы все это по-прежнему можете – если очень хочется или почему-либо нужно. Но в томто и дело – это самое «нужно» возникает теперь крайне редко.

Дистрибутив типа Ubuntu/Kubuntu разворачивается до полностью работоспособного состояния за полчаса и не требует никаких дополнительных действий (кроме разве что сугубо косметических). Не намного больше времени уйдет на установку Archlinux или Zenwalk, которые не считаются эталонами дружественности и тем не менее требуют только многократно документированной правки пары-тройки конфигурационных файлов. Правда, все это – при условии хорошего доступа к Сети.

У вас проблемы со связью? И это не смертельно, Mandriva, например, может быть полностью укомплектована без доступа в Интернет вообще. Вероятно, то же самое относится к SUSE, Fedora, клонам Red Hat: я говорю только о тех системах, которые «щупал» в последнее время. И писать о настройках системы ныне становится просто смешно.

Так что же, братья-линуксописатели, пора переквалифицироваться в управдомы? Отнюдь. Наступает время, когда можно спокойно начинать писать о том, что относится к непосредственной сфере деятельности. Не о том ли мечтали мы долгие годы?

alv@posix.ru

Сегодня мы рассматриваем...

Gentoo с. 14



› «Нет! Не нажимай на красную кнопку!» Порой во время инсталляции чувствуешь себя Бондом...

Ballistics с. 18



› Нашему обозревателю пришлось сильно притормозить, чтобы сделать этот скриншот для вас. Вот как бывает...

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth

Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.

Рейтинг 9/10

Yoggie Gatekeeper Pro

Yoggie ставит Linux между ноутбуком и Сетью. Эта крошечная штучка легко влезает в карман, и битком набита защитными программами. Пробует **Грэм Моррисон**.

Вкратце...

» Маленькое встраиваемое Linux-устройство, используемое в качестве шлюза безопасности. Превосходно для защиты удаленных сетевых подключений. См. также: А смотреть нечего. Лишь программное решение может обеспечить такую портативность.

Если вы недавно видали Microsoft Windows, то, вероятно, заметили избыток защитного ПО, работающего на компьютере во время походов по Интернет. По самым скромным подсчетам, необходим брандмауэр, антивирусный сканер и какая-нибудь защита от троянов. Все эти приложения съедают изрядную долю производительности процессора.

Разработчики Yoggie считают, что нашли подходящее решение – поручить все эти трудоемкие задачи отдельному компьютеру, дезинфицирующему все поступившее из сети, не давая ему шансов навредить. Это решение воплотилось во встраиваемое Linux-устройство, которое легко помещается на ладони. Под его черно-серой лоснящейся пластиковой поверхностью скрыты процессор Intel PCA 270 520 МГц и 128 МБ ОЗУ, обладающие мощностью Pentium'a III из далекого 1999 г. Но, в отличие от 1999, этот процессор выполняет стандартную версию BusyBox и Linux 2.6. Все это скрыто от глаз, поскольку единственное предназначение Yoggie – стоять на страже безопасности.

Разворачиваем USB-кабель

Перед включением устройства нужно размотать плоский USB-кабель, удобно обернутый вокруг пластикового корпуса. Если у вас Microsoft Windows XP или Vista, то USB-порт – все, что вам нужно. Подключите устройство к компьютеру и установите драйверы. Yoggie поместит свой драйвер в самую глубь сетевого слоя, фильтруя данные в обоих направлениях до того, как они попадут на основную машину.



Linux-пользователям понадобится аппаратный эквивалент: два разъема RJ45, скрытые под USB-кабелем, образуют мост между Интернетом и вашей машиной. Это означает, что пользоваться встроенной беспроводной картой под Linux вы не сможете, хотя специальный Linux-драйвер для этого уже обещан.

Однако конечный результат не зависит от используемой ОС: Windows или Linux. Необработанные сетевые данные поступают в устройство и, прежде чем попасть в основной компьютер, фильтруются сквозь 12 преднастроенных защитных программ. Среди этих программ – детектор вторжения, брандмауэр, VPN, фильтр контента и антивирус Касперского, в основном используются прозрачные прокси для HTTP, POP и SNMP. Антивирус, на наш взгляд, вполне работоспособен, умеет находить вирусы в упакованных вложениях, но вот вирусы, передаваемые через HTTPS, отловить не сможет, а ведь по этому протоколу в основном получают почту через web-интерфейс.

Включи и забудь

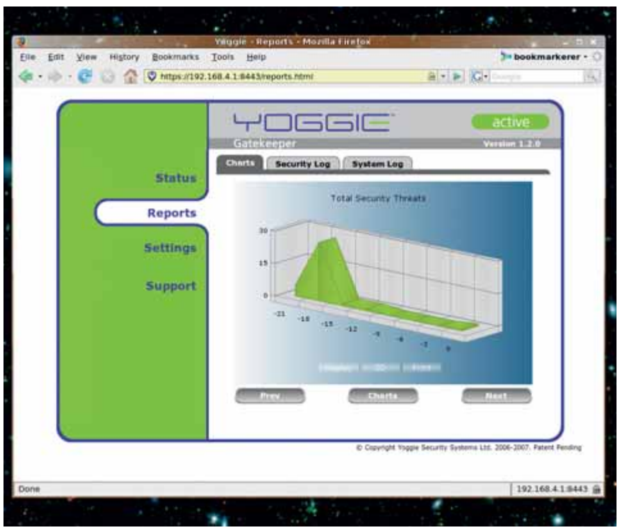
Linux-пользователю понадобится подключить и USB-кабель, чтобы подать на устройство питание, а после этого подождать около 90 сек для загрузки Yoggie. Обещано обновление прошивки, которое сократит это время вдвое, но на самом деле это не суть важно, если только вы не рветесь очертя голову получить заветную почту. Когда синие индикаторы на передней панели устройства перестанут мигать, можно выходить в Интернет.

Устройство настраивается через опрятный и удобный web-интерфейс, и первое, что надо сделать – ввести лицензионный ключ, идущий в комплекте. Yoggie постоянно обновляется с загрузкой новых антивирусных баз и прошивок, в цену продукта включена стоимость годового обслуживания. В течение суток наши системные журналы зарегистрировали троекратное обновление антивирусных баз. Для графического отображения работы устройства широко используется Flash-анимация, каждая прореха в системе безопасности регистрируется и выводится на панель.

› Устали разбираться с версиями защитных программ от разных поставщиков? Получайте эффективное решение типа «все-в-одном»!

По сравнению со среднестатистическим Linux, настройка политики безопасности Yoggie проста. IT-менеджер может даже приобрести у Yoggie Linux-сервер, чтобы распространить одну и ту же политику на целую эскадру ноутбуков: это отличный способ защитить свою сеть при удаленном доступе.

Короче, Yoggie – это успех. Устройство недорогое, купить его сможет даже постоянный пассажир авиалинии эконом-класса; и оно обеспечивает тот уровень душевного комфорта, которым может побаловать только Linux. LXF



› Для решения административных задач вроде настройки профилей безопасности и наблюдения за состоянием безопасности служит web-интерфейс.

LINUX FORMAT Вердикт

Yoggie Gatekeeper Pro

Производитель: Yoggie Security Systems
Сайт: www.yoggie.com
Цена: $220

Функциональность	8/10
Производительность	7/10
Простота использования	8/10
Цена	9/10

» Обеспечивает безопасность даже в дороге, но дополнительный антивирус все же необходим.

Рейтинг 8/10

Studio To Go 2

Грэм Моррисон нашел сверх-оснащенную студию звукозаписи на CD, которую можно всегда носить с собой в ожидании музыкального вдохновения. Везет!

Вкратце...

» Живой Linux-дистрибутив, битком набитый новейшим аудио- и музыкальным ПО. Среди свободных альтернатив: Ubuntu Studio, Agnula и 64 Studio.

Linux – прекрасный выбор для меломана: здесь так много интересных и разных музыкальных приложений, что ваша музыка на другой ОС, вероятно, никогда не достигнет такого звучания. Проблема одна: для правильной работы системы необходимы некоторые навыки работы в Linux – для среднего музыканта непростая задача. К счастью, есть Studio To Go, пакет лучших аудио-приложений для Linux, уместившийся на единственном CD, заранее усовершенствованный, преднастроенный и предназначенный для работы с большинством компьютерного и звукового оборудования.

Первую версию Studio To Go мы рассматривали в LXF70; миновало два года, и после долгого бета-тестирования вышла, наконец, Studio To Go 2. За это время многое изменилось в разработке ПО для аудио под Linux, и STG 2 должна соответствовать последним веяниям в области эффектов реального времени и аудио-синтеза, чтобы удержать лидерство, а заодно и оправдать довольно высокую цену.



› ***Studio To Go 2* втиснул на единственный диск все известные нам аудио-приложения. Если чего-то здесь нет – значит, этого еще не придумали.**

> «Лучшая подборка “родных” Linux-модулей из виденных нами.»

Первое, что бросается в глаза – улучшенная поддержка оборудования: большее количество профессиональных аудиокарт стало работать без дополнительной настройки. У нас не возникло проблем ни со встроенным аудиочипсетом, ни с M-Audio Audiophile 2496. При первом появлении рабочего стола вас встретит уведомление о том, какая аудиокарта была настроена. Данная версия поддерживает даже различные USB- и FireWire-устройства, благодаря наличию *Freebob*, свободной комбинации прошивки и драйверов для внешних устройств. Чтобы убедиться в совместимости с вашей аппаратурой, можно попробовать демо-версию STG 2.

Сердце системы – впечатляющий *Rosegarden*, лучший из известных на Linux аудио/MIDI секвенсоров (мы рассматривали версию 1.5 в LXF92). Один из основных разработчиков *Rosegarden*, Крис Кэннем [Chris Cannam], заодно принадлежит и к тем парням, что стоят за Studio To Go 2.

Синтез идей

Отличительные черты STG 2 – эффекты и синтез. Наиболее интересная добавка – ссылка на папку рабочего стола под названием VST Plugins. *VST* (*Virtual Studio Technology*) от Steinberg – это основа эффектов реального времени и синтеза для Windows и OS X, именно она вдохновила революцию в звукозаписи, усеяв eBay устаревшей аппаратурой: программные эквиваленты ее вытеснили. Щелчок на одной из библиотек в папке *VST* вызывает *Wine* (поскольку библиотеки пришли из Windows) и автоматически встраивает вызванный модуль в аудио-сервер *Jack*. После этого можно конвейером передавать аудио в любое *Jack*-совместимое аудио-приложение и получать его обратно (Jack–совместимо почти все).

STG 2 включает также лучшую из виденных нами подборку «родных» Linux-модулей, и это превосходно иллюстрирует, насколько продвинулось создание эффектов реального времени на нашей любимой ОС. Взгляните хотя бы на синтезаторный эмулятор *Bristol* или набор эффектов Creox – и поймете, что мы имеем в виду.

Как Linux-пользователи, мы привыкли пользоваться бесплатным ПО. Но в мире написания музыки и звукозаписи пакет, обеспечивающий аудио/MIDI-секвенсирование, лучшие из известных аудиоэффектов, аудио-редактирование, запись CD, сэмплинг, синтезирование и спектральный анализ, может стоить запредельно, так что цена под $160 вполне оправдана. Во многих отношениях STG 2 занимает лидирующие позиции и обладает таким количеством прекрасно интегрированных нововведений, что непонятно даже, с чего начать. LXF

Лучшее ПО

Приложения
- » *Rosegarden 1.5.5*
- » *Ardour 2.0.2*
- » *Sonic Visualiser 1.0*
- » *Free Music Instrument Tuner 0.96.7*
- » *Jamin 0.95*

Синтезаторы
- » *Fluidsynth 1.0.7*
- » *Hydrogen 0.9.3*
- » *Aeolus 0.6.6*
- » *Bristol 0.9.5*
- » *LMMS 0.2.1*

Эффекты
- » *FreeWheeling 0.5.5*
- » *Freqtweak 0.6.1*
- » *SooperLooper 1.0.8*
- » *Ecamegapedal 0.4.4*
- » *TerminatorX 3.82*

Linux Format Вердикт

Studio To Go 2

Разработчик: Fervent Software
Сайт: www.studio-to-go.com
Цена: £79.99 (ок. $160)

Функциональность	10/10
Производительность	8/10
Простота использования	7/10
Цена	7/10

» *Могучий пакет королевского размаха. Позаботьтесь о просторных апартаментах для него на своем жестком диске.*

Рейтинг 8/10

Gentoo 2007.0

Можно ли инсталлировать Gentoo, не имея ученой степени? **Нейл Ботвик** пробует, хотя диссертаций не защищал.

Вкратце...

» Метадистрибутив на базе исходных текстов, предоставляющий пользователю полный контроль над системой. См. также: Linux from Scratch или Sabayon Linux.



Пол считает...

«Версия 2007.0? Помилуйте, я еще не закончил компиляцию 2006.0! Ну да ладно, по крайней мере, графический инсталлятор больше не глючит…»

Gentoo пользуется репутацией запредельного дистрибутива, интересного лишь охотникам за скоростью, выжимающим производительность из каждого цикла процессора до капли. Так ли это? Мы инсталлировали новейшую версию.

Конечно, особый метод инсталляции ПО Gentoo никуда не делся и в версии 2007.0. В отличие от большинства дистрибутивов, здесь пользователь во время инсталляции должен компилировать все программы из исходных текстов, а не устанавливать типовые пакеты, подготовленные разработчиками. Компиляция из исходников может показаться весьма сложной, но система управления пакетами *Portage* берет на себя заботу о деталях. Укажите имя пакета, который хотите установить, и она загрузит код со всеми его зависимостями, скомпилирует и инсталлирует готовую программу. В целом процесс не сложнее, чем установка пакетов на Debian, хотя, бесспорно, гораздо дольше.

Преимущество такого метода состоит в том, что вы получаете полный контроль над инсталляцией. Именно этот контроль, а не использование заумных флагов компилятора, придает Gentoo особую силу. USE-флагами отмечаются те свойства ПО, которые вы намерены использовать; удаляя ненужные, вы экономите место, избавляетесь от лишних зависимостей, сокращаете время загрузки программ и даже снижаете количество брешей в системе безопасности.

Сборка собственного ядра также упростилась благодаря инструменту *genkernel*, которым пользуется инсталлятор; его можно применять и для компиляции ядра вручную. В запасниках Portage содержится около 12 000 пакетов, поэтому голодный паек вам не грозит.



› Графический инсталлятор теперь надежен и готов помочь на каждом этапе процесса установки.

Улучшения инсталлятора

По сравнению с прежними версиями установка Gentoo существенно упростилась. Для тех, кто желает знать, как работает система, оставлена прежняя текстовая процедура инсталляции, ускоренная наличием двоичных пакетов Стадии 3 (Stage 3) на инсталляционном диске. Но графическим инсталлятором пользоваться гораздо удобнее. Он был представлен еще в 2006.0, но страдал недомоганиями бета-статуса. Теперь все намного лучше: есть GTK-интерфейс, предоставляющий на выбор все доступные параметры при разумных умолчаниях. Добавлена функция локальной инсталляции – она работает так же, как и в большинстве других дистрибутивов, позволяя установить все доступное ПО с диска. Стандартный процесс инсталляции дает больше свободы, позволяя взять все необходимое из Интернета. Локальная инсталляция, конечно, быстрее, ибо загруженные пакеты нужно еще скомпилировать.

Продвинутый режим инсталляции дает безграничные возможности настроить (или угробить) вашу систему; есть и текстовый вариант. Этот режим может пригодиться на слабом компьютере, особенно при сетевой инсталляции. Есть возможность сохранять и загружать инсталляционный профиль для клонирования конфигурации на нескольких компьютерах.

Локальная инсталляция включает Gnome 2.16.2, KDE 3.5.5 и *OpenOffice.org 2.1.0*. Изучить Gentoo намного сложнее, чем другие дистрибутивы, но им проще управлять, благодаря дополнительному уровню контроля. Превосходная документация оказывает в этом неоценимую помощь. Но особый контроль означает и особую ответственность пользователя за свои действия – получив возможность усовершенствовать свою систему, вы получаете и шанс разрушить ее в таких изощренных формах, что пользователям Ubuntu даже и не снились.

Как можно сравнить Gentoo с другими дистрибутивами? Да никак! Только Linux From Scratch даст подобный уровень контроля, но при значительно больших усилиях. Часто вспоминают Sabayon Linux, ведь он основан на Gentoo. Но Sabayon – скорее иллюстрация возможностей Gentoo, чем реальный соперник.

Если вы хотите полностью контролировать свою систему и готовы для этого потрудиться, то Gentoo – это ваш выбор. Тому, кто привык вставлять установочный диск и сразу же отправляться на просторы Сети, лучше будет взять обычный двоичный дистрибутив. LXF

Стадия 1, 2 или 3?

Прежде инсталляция Gentoo представляла собой последовательность стадий. Стадия 1 включала все необходимое для загрузки системы, все остальное собиралось из исходников. Стадия 3 полностью устанавливала базовую систему из прекомпилированных пакетов, что существенно сокращало время установки. Стадия 2 была чем-то промежуточным. Теперь Стадия 3 стала стандартным (и единственным поддерживаемым) методом, поэтому время инсталляции Gentoo можно измерять чашками кофе, а не оборотами часовой стрелки.

На DVD теперь есть пакеты GRP (Gentoo Reference Platform), двоичные пакеты ПО общего назначения.

LINUX FORMAT Вердикт

Gentoo Linux 2007.0

Разработчик: Gentoo Foundation
Сайт: www.gentoo.org
Цена: Бесплатно, в основном под свободными лицензиями

Функциональность	8/10
Производительность	9/10
Простота использования	7/10
Документация	10/10

» *Превосходный выбор для тех, кто готов потрудиться ради ОС, скроенной по своей.*

Рейтинг 9/10

Технология счастья

SUNRADIO.RU

сетевое радио под ключ на базе Linux

новое будущее вашей компании

pr@sunradio.ru +7 812 955 76 70 www.sunradio.ru

LINUX FORMAT на DVD

» AstroMenace 1.1 и 1.2 (тестовая)

AstroMenace

Может ли простая видеоигра выбить из рабочего ритма на несколько дней? Давний противник электронных развлечений **Евгений Балдин** еле оторвался от монитора, чтобы дать ответ на этот вопрос.

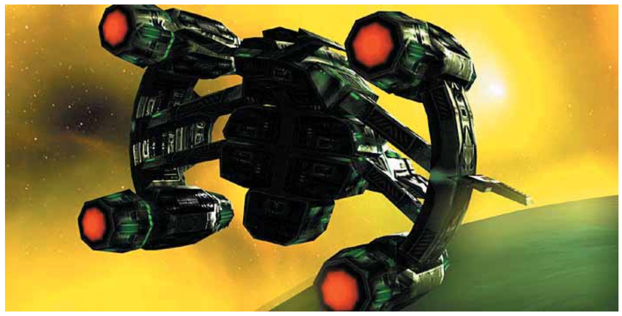
Вкратце...

» Фантастическая вертикальная «стрелялка» с легко настраиваемым уровнем сложности, а так же, возможно, излишне простой экономической моделью и с элементами реальных физических моделями полета и наведения оружия. См. также: *xsoldier*, *XGalaga*.

Вертикальная «стрелялка» – всего два слова и почти все необходимые ассоциации уже возникают сами собой. В вашем распоряжении космический корабль, горы оружия, тучи врагов и задача как можно эффективнее наводить это оружие на вражеские истребители, бомбардировщики, ракетоносцы, танки, стационарные зенитки и тому подобное.

Пятнадцать миссий, фактически отсутствующий сценарий, десятки разнообразных красиво вылепленных противников, множество моделей оружия и спецэффектов от их применения – присутствуют все элементы жанра. В качестве минимальной конфигурации декларируется CPU 1+ ГГц (при этом уточняется, что имеется в виду «честный», а не урезанный процессор типа Celeron/Duron), 128 МБ ОЗУ, видеокарта с 3D-ускорителем (OpenGL) и более 32 МБ памяти (опять же оговаривается что этот минимум ориентируется на внешние, а не встроенные видеокарты) и, естественно, мышь.

В игре реализуется «честная» трехмерия. Любой из 125 объектов можно рассмотреть в разделе «Информация» основного меню игры и изучить его параметры. Сама по себе стрелялка горизонтальная, но рассматривается как бы сверху, и враги могут лететь на разной высоте по отношению к вашему космическому кораблю. По утверждению автора, вначале закладывалась возможность поворачивать камеру обзора, но ее реализацию в полном объеме пока решили отложить. То, что противник может оказаться на иной высоте, нежели игрок, усложняет процесс его уничтожения. Одной из ключевых особенностей игры является физическая модель, по которой происходит наведение оружия: в зависимости от уровня установленного на корабле игрока компьютера и системы наведения уничтожение противника происходит с разной степенью эффективности. Особенно эти параметры влияют на ракетное оружие.





Упрощенная экономическая модель игры основывается на ликвидации кораблей противника: чем больше сбил – тем больше денег. Деньги начисляются за каждого уничтоженного врага. В любой момент можно пройти любую миссию еще раз, но при этом размер премии за подбитого противника уменьшается вдвое по сравнению с предыдущей попыткой. Все накопленные деньги в виде закупленного железа и оружия в любой момент можно трансформировать в более эффективный боевой корабль и вооружение. Игрок может примерить разные типы кораблей – от легкого разведчика до тяжелого бомбардировщика. С одной стороны, нет опасности неправильного «развития» – всегда можно переиграть, а с другой – уходит один из важных элементов игры, связанный с планированием и обустройством своего космического штурмовика.

Построение миссий от простого к сложному вынуждает вас покупать более мощные корабли, ставить на них более тяжелое вооружение и защитные системы, что в свою очередь подразумевает более мощную энергетическую установку. Денег всегда не хватает и в какой-то момент миссия может оказаться невыполнимой – тогда можно легко и непринужденно изменить уровень сложности в меню выбора профиля пилота, например, увеличив эффективность своего оружия и уменьшив эффективность оружия противника или отключив физическую модель наведения, предпочтя ей аркадную. Уровень сложности прямо пропорционально влияет на величину денежной премии за убитых врагов, зато позволяет пройти сложные уровни без напряжения и мучительного выбора подходящей комбинации корабль/оружие. Иногда достаточно просто уменьшить скорость протекания игры. Замедлить или ускорить время можно прямо во время

Кстати

AstroMenace 1.2 является тестовой версией, которая будет интересна владельцам широкоформатных мониторов.







AstroMenace



› **Огонь! И да пребудет с нами сила!**

миссии, нажимая клавиши F5 и F7, то есть стало сложно – замедлился в десять раз, надоело – ускорился в пять. Будьте аккуратнее с клавишей F6 – она сбрасывает скорость до значения по умолчанию.

В дополнение к самой игре команда разработчиков предлагает продвинутым пользователям возможность создавать свои собственные уровни. Редактор уровней отсутствует (в силу особенностей жанра в нем нет необходимости), зато есть подробное описание текстового формата файлов миссии. В завершение обзора следует отметить, что у игры есть свое, хоть пока и небольшое, но довольно активное русскоязычное сообщество, которое влияет на политику добавления тех или иных особенностей и новых возможностей *AstroMenace*. LXF



› **На горизонте… Земля!!!**

LINUX FORMAT Вердикт

AstroMenace

Разработчик: Viewizard Games
Сайт: www.viewizard.com
Цена: Бесплатно для пользователей GNU/Linux

Сюжет	2/10
Графика	9/10
Увлекательность	8/10
Цена	9/10

» *Скажем просто: затягивает.*

Рейтинг 8/10

Ballistics

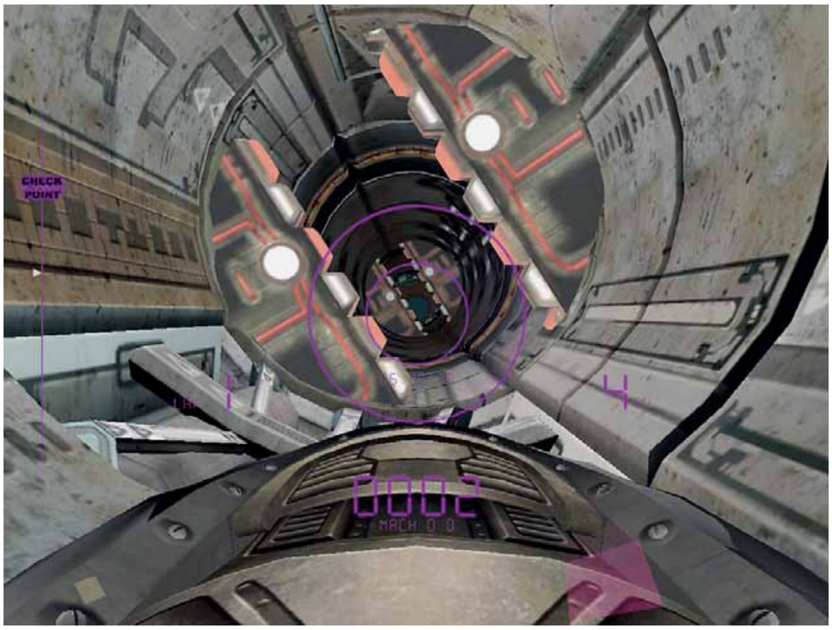
Penguin Racer не балует адреналином? **Грэм Моррисон** нашел кое-что покруче.

Вкратце...

» Высокоскоростная футуристическая культовая игра. Альтернативы на Linux просто нет, Penguin Racer и рядом не стояла.

Linux Game Publishing разыскала игру, достойную возрождения. *Ballistics* впервые увидела свет пять лет назад, на Microsoft Windows и достопочтенном Xbox. Но она до сих пор остается в игротеках внушительного числа игроков, создав нечто вроде культа среди любителей быстрой езды. Как следует уже из названия, в *Ballistics* все подчинено скорости – питаемой кофеином, доводящей до головной боли, рвущей нервы скорости.

Прежде чем подойти к первой звуковой, оглядимся вокруг. Мы в 2090-м году, и человечество уже не стоит на краю пропасти. Войны и эпидемии ушли в прошлое, почти не оставив среднему гражданину серьезных поводов для беспокойства. Вместо амфетаминов и апатии, люди 2090 подсели на более естественный наркотик: адреналин. Чрезвычайно популярны экстремальные виды спорта, а гонки '*Ballistics*' среди них – последний писк.



› **Проложить свой путь непросто: коридор сужается, скорость растет.**

Выбираем снаряд

Основа *Ballistics* – подобие спортивного мотоцикла XXII века. Похожие на скоростные мотоциклы без колес или Light Cycle из *Tron*, они гоняются в широких подземных трубах, змеящихся по разным экзотическим местностям, подчас достигая скорости свыше 1000 км/ч.

Пилоты этих магнитных снарядов годами тренируются только для того, чтобы принять участие в чемпионате. Их реакция быстрее полета пули, нашпигованной электронными имплантатами. Даже неловко, ведь их судьба доверена линуксоидам, которые управляют болидами 2090 года с помощью устройства, запатентованного Кристофером Шоулзом [Christopher Sholes] в 1898 (та самая Qwerty-клавиатура, что перед вами). Ваша задача – победить, не имея за спиной опыта многолетних тренировок, и по возможности набрать достаточно денег и удачи, чтобы ваш персонаж удалился на заслуженный отдых, а не на вечный покой.

Многолетние упражнения заменяет тренировочный режим, который научит вас управляться со скоростью и посоветует несколько выигрышных стратегий. Голосовые подсказки помогут освоиться со снарядом, правильно расположить пальцы и грамотно пройти первый поворот.

Есть два режима передвижения. Стандартный, самый быстрый – когда снаряд магнитом притягивается к стенке трубы. Левая

Шаг за шагом: Изучение техники гонок



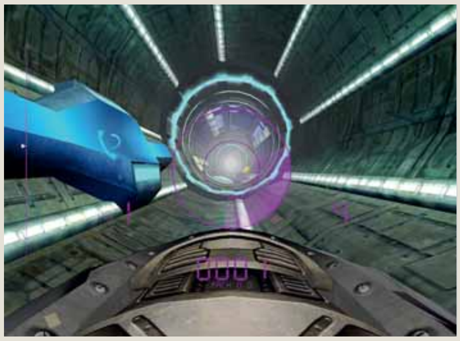
1 Мощность

Оторвитесь от стенки трубы, чтобы сцапать дополнительные очки мощности. Для продолжения гонки, прижмитесь обратно.



2 Сила

Проехав по желтым панелям, вы обретете дополнительные силовые резервы. Нажмите и удерживайте клавишу пробела.



3 Ремонт повреждений

Синие панели охлаждают двигатель и устраняют повреждения.

кнопка мыши постепенно ускоряет движение, а клавишами D и F скоростник смещается влево и вправо, как и в других играх. Можно даже навсегда залепить левую кнопку мыши в нажатом состоянии, если вы не собираетесь убирать с нее палец. Правая кнопка, напротив, используется очень редко. Ее нажатие устраняет магнитное притяжение, прижимающее снаряд к стенке туннеля, и предоставляет ему беспомощно болтаться в воздухе посреди трубы. Теперь болид управляется с помощью мыши, но быстро теряет скорость: ускорение возможно лишь тогда, когда он прижат к поверхности.

Однако парение в середине трубы – это единственное средство избежать некоторых препятствий и набрать предлагаемые по пути очки повышения мощности. Есть три разных вида таких очков: денежные, охлаждающие и силовые. Повторное нажатие правой кнопки притянет аппарат к ближайшей поверхности.

В трубах встречаются различные объекты, загромождающие путь. Все, что окрашено красным, необходимо объезжать: столкновение приводит к повреждению и, хуже того, полной потере скорости. Но есть и полезные объекты. Желтые панели наполнят снаряд энергией, которую можно использовать, нажав клавишу пробела. Синие панели охлаждают двигатель и постепенно исправляют повреждения. Научиться управлению нетрудно, и очень скоро вы сможете участвовать в чемпионате.

По ту сторону бесконечности

Известность и удача зависят от побед. В чемпионате это означает выигрыш нескольких гонок подряд, на разных трассах. Чтобы пройти квалификацию, нужно попасть в первую тройку, затем переходить на другую трассу.

За каждую победу вы получаете виртуальные деньги. Их необходимо потратить на усовершенствование своего аппарата, чтобы он быстрее передвигался, маневрировал и мог переносить более серьезные повреждения. Состязания на высшем уровне невозможны без таких обновлений. Всего трасс семь, а для квалификации дается только три попытки. Это придает гонкам азарт.



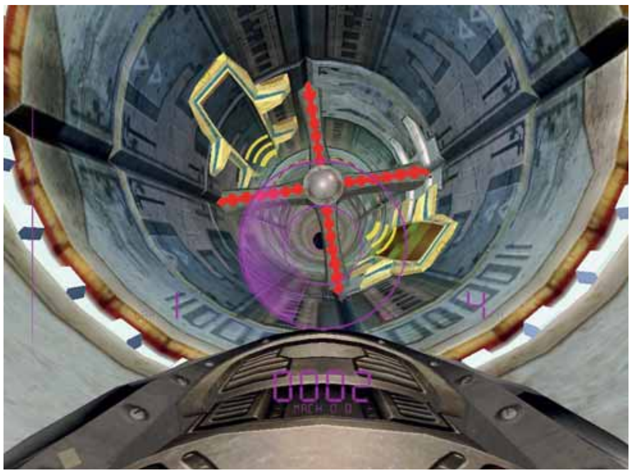
› Большинство трасс подземные, с редкими выходами на поверхность. Пейзажем тут любоваться некогда.

Езда чем-то напоминает катание на американских горках. По мере набора скорости стенки туннеля надвигаются на вас все быстрее. На старте еще можно различить мелкие детали на боковых панелях: футуристические вентиляционные решетки, отводные трубы и окна. В зависимости от места проведения гонки, в вашем окне будет мелькать Большой Каньон, просторы Аляски или небоскребы Токио. Текстуры весьма несовершенны: игре все-таки пять лет. Зато для нее не обязательно иметь суперкомпьютер.

> «На скорости 1000 км/ч видны размытые линии, а не детали — вы в Зоне.»

Временами туннель вырывается на поверхность и взмывает вверх, затем снова уходит под землю. Перепады высот достаточны, чтобы создать дополнительную сложность. Но окружающая обстановка скоро начинает размываться. На скорости 400 км/ч еще можно разглядеть достаточно деталей, чтобы понять, где вы находитесь, предвидеть появление препятствия и шарахнуться влево или вправо для его обхода. На скорости свыше 400 км/ч окружение приобретает более абстрактную форму. Пальцы мечутся то влево, то вправо, а поле зрения вытягивается. На 1000 км/ч деталей уже не увидишь, лишь размытые линии да дисплей, светящийся на лобовом стекле – вы в Зоне. Преодолев порог 1600 км/ч, вы получите вознаграждение в игровой валюте; разогнавшись до скорости свыше 2000 км/ч, остается лишь довериться своим инстинктам. Зачастую вы заканчиваете гонку, ощущая одновременно и эйфорию, и изнурение.

Вы любо влюбитесь в эту игру, либо ее возненавидите. Но такого ощущения скорости мы еще никогда не испытывали: например, *Wipeout* гораздо медленнее. С каждой неудачей вы все сильнее рветесь назад, в гонку, а онлайн-режим нескольких игроков добавляет еще одно измерение. Здесь не обойтись без сверхбыстрой реакции и устойчивости к стробоскопическим мельканиям, но взамен вы приобщитесь к одной из лучших гоночных игр, доступных под Linux. LXF



› Все, что помечено красным, надо обходить: столкновение повредит ваш болид, и, хуже того, он потеряет скорость.

LINUX FORMAT Вердикт

Ballistics

Разработчик: Linux Game Publishing
Сайт: www.linuxgamepublishing.com
Цена: £28 (ок. $55)

Сюжет	9/10
Графика	7/10
Увлекательность	8/10
Цена	7/10

» Добиться таких же ощущений от Penguin Racer *можно, только обкормив допингом добрую тысячу пингвинов.*

Рейтинг 8/10

Сравнение

» Каждый месяц мы сравниваем тонны ПО – а вы можете отдыхать!

Почтовые клиенты

Что лучше: полный канцелярский набор *Thunderbird*, кожаная перочистка *Opera* или бумага с карандашом от *Mutt*? Путь к контактам выбирает **Крис Хоуэллс**.



Paul Blachford

Про наш тест...

Мы испытывали почтовые клиенты на Kubuntu 7.04 на ноутбуке Lenovo ThinkPad X60s, с 1,5 ГБ ОЗУ и 1,66-ГГц процессором Intel Core Duo, ограниченным одним ядром.

» **Безопасность и секретность** Мы смотрели, имеет ли клиент поддержку шифрованных и подписанных почтовых сообщений, а также какую-нибудь помощь в распознавании фишинга. Мы также исследовали противоспамные средства.

» **Функциональные возможности** Мы рассматривали протоколы, поддерживаемые каждым клиентом, и число доступных полезных свойств.

» **Простота в использовании** Можно ли давать такую программу неопытным пользователям, и надеяться, что ею будут пользоваться, прибегая к помощи лишь изредка.

» **Скорость и эффективность** Мы измеряли время, требуемое для загрузки почтового ящика с 60 000 письмами, а также поиск строки “christmas” в теле и в теме письма, затем брали среднее этих трех величин. Также обращалось внимание на то, ухудшается ли время загрузки для больших почтовых ящиков.

Наш выбор

Немногое произвело такую же революцию в области связи в XXI веке, как электронная почта. К сожалению, по мере того, как взрослела электронная почта, росло и умение любителей злоупотребить ею. Ненавистная Незатребованная Коммерческая Почта (Unsolicited Commercial Email, UCE), или спам, продолжает развиваться. По оценкам Messaging Anti-Abuse Working Group (Группа по борьбе со злоупотреблением почтой, www.maawg.org), в последнем квартале 2005 года 80–85% входящего почтового трафика составляла «непрошеная почта» – это где-то в районе 90 миллиардов писем в день!

Еще опаснее фишинг, технология втирания в доверие, направленная на вытягивание секретной информации у наивных жертв. Вирусы и трояны обычно не беспокоят пользователей Linux, поскольку число их невелико, однако вполне возможно употребить пользователя Linux как вектор атаки, направленный на Windows-машины, при беззаботной рассылке прикреплений.

Почтовые клиенты предусматривают защиту от этих угроз, однако ваш фактор выбора клиента обычно один: вызывает ли процесс работы с электронной почтой удовольствие или раздражение? Наше **Сравнение** попытается выявить лучшего.

Как работают почтовые клиенты?

Как вы, наверное, знаете, в основном используются два способа сбора электронной почты. Это POP (Post Office Protocol) и IMAP (Internet Message Access Protocol). В протоколе POP текстовое содержимое и все прикрепления скачиваются с удаленного почтового сервера на локальный компьютер, затем немедленно удаляются на почтовом сервере [это не является требованием протокола, но так обычно и происходит, – прим.ред.]. С этого момента ответственность почтового сервера за электронную почту заканчивается.

IMAP, напротив, хранит все сообщения на сервере, и требует постоянной связи по сети или через Интернет. Это может раздражать, если вы путешествуете и имеете ограниченный доступ к Интернету, хотя существует несколько почтовых клиентов, способных скачивать письма IMAP для офлайнового использования. Наши клиенты поддерживают оба метода.

Это были азы – теперь двинемся дальше!

Opera Mail

Уважаемый и многофункциональный проприетарный клиент.

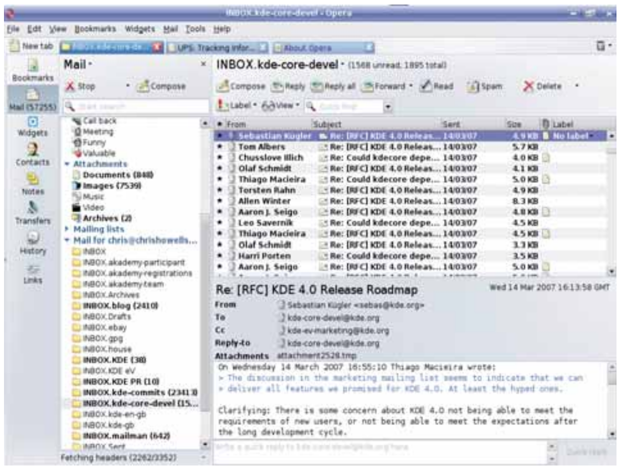
Начав просто как браузер, с годами *Opera* превратилась в обширный комплект программ, куда входит и почтовый клиент. Он поддерживает и POP3, и IMAP, и позволяет применять SSL-шифрование для улучшения безопасности этих протоколов. Также поддерживается SMTP-аутентификация, что позволяет вам подтверждать подлинность на удаленном почтовом сервере. Вызывает некоторое раздражение, что установленная по умолчанию подпись рекламирует продукцию *Opera*, но ее легко убрать.

Opera создает несколько виртуальных папок, основанных на содержимом вашей электронной почты. Внутри виртуальной папки Attachments (Вложения) вы найдете подкаталоги для документов, рисунков, музыки, видео и сжатых архивов (например, ZIP-файлов). Можно выбрать любой из этих подкаталогов, чтобы увидеть все письма, отвечающие данному признаку.

Имеется хорошая поддержка для почтовых списков рассылки, и *Opera* распознает электронную почту, которая распространяется утилитами рассылок, например, *GNU Mailman*. Она организует эти письма в другие виртуальные папки: не имеет значения, в какой именно папке хранится письмо, можно отображать все письма из определенного списка. Для легкой категоризации, опять-таки путем виртуальных папок, письмам можно присвоить метки-теги.

Нам приглянулся вид функции Quick Reply (Быстрый ответ), появляющейся внизу каждого письма. Она позволяет отвечать на письма, не открывая отдельное окно для написания текста. К сожалению, на практике оказалось, что поддерживается только топ-постинг [т.е. текст ответа помещается над цитированием оригинального письма, – прим. ред.].

В целом, производительность *Opera* выглядит неравномерной. Она была очень медленной при импортировании нашего тестового почтового ящика, потребляя бóльшую долю времени CPU. Это может доставить неприятности, так как во время проверки почтового ящика, как правило, занимаются и другими делами. После этого производительность была потрясающей, загрузка ящика занимала лишь 0,5 секунды, а поиск в теме и содержимом письма был мгновенным.



› **Имея вид вкладки главного окна браузера *Opera*, почтовый клиент *Opera* легко вызывается из других приложений пакета.**

LINUX FORMAT Вердикт

Opera

Версия: 9.20
Сайт: www.opera.com
Цена: Бесплатно, поддержка за дополнительную плату

» *Большое число новых функций с продуманным оформлением превращают использование почтового клиента Opera в удовольствие..*

Рейтинг 7/10

Mutt

В отличие от *Opera*, минималистическое приложение.

В нашем обзоре *Mutt* стоит особняком, поскольку это здесь единственное консольное приложение. Первоначально основанный на старом почтовом клиенте *Elm*, *Mutt* с тех пор стал довольно мощным приложением.

Откровенно говоря, *Mutt* имеет довольно крутую «кривую обучения» по сравнению со многими другими программами, однако на wiki-страницах сайта громоздятся горы документации. Его текстовая природа даст много преимуществ по сравнению с графическими альтернативами, если вы ухватите ее суть. Например, вы легко сможете проверить вашу почту отовсюду, используя SSH для соединения с сервером, где установлен *Mutt*, а затем запустить программу, как если бы вы использовали ее локально.

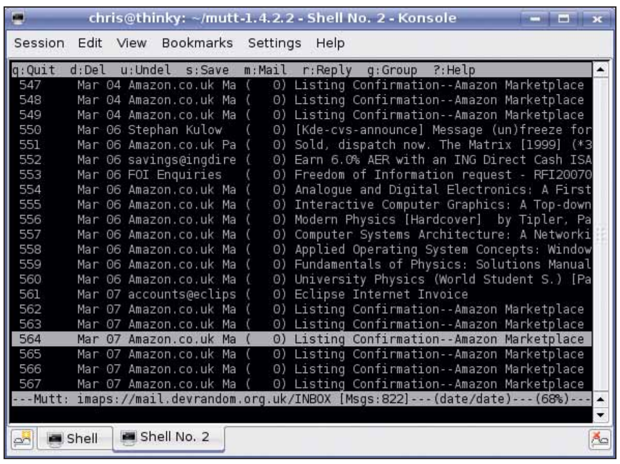
Mutt управляется исключительно клавиатурой. По умолчанию дается большое число горячих клавиш, а настроить можно еще больше. Также есть возможность использовать макросы, и они работают, выполняя последовательность нескольких функций вместе за один шаг.

Вы можете настроить клиента самостоятельно путем ручного редактирования файла **.muttrc**. Файлы **.muttrc** целиком доступны на сайте *Mutt* в качестве примеров, а в руководстве по *Mutt* можно найти фрагменты опций настройки, пригодные для вставки в файл конфигурации, чтобы настроить *Mutt* на ваш вкус.

Предусмотрен встроенный редактор для написания писем, однако если вы предпочитаете внешний текстовый редактор вроде *Vi* или *Emacs*, переключитесь на них – либо посредством изменения **.muttrc**, либо путем задания переменной среды $EDITOR. По умолчанию *Mutt* не может просматривать HTML или создавать HTML-письма. Желающим видеть HTML-письма придется настроить *Mutt* на использование внешнего HTML-просмотрщика/браузера, например, *Lynx*.

Производительность *Mutt* не была выдающейся. Загрузка тестового ящика заняла 10,3 секунды. Поиск по письмам прошел за 12,5 секунд, а вот по темам писем – за более приятные 0,8 секунды.

> «Если вы предпочитаете редактор *Vi* или *Emacs*, можно встроить его в *Mutt*.»



› **Пользовательский интерфейс *Mutt* – на базе *Ncurses* или S-Lang, и можно даже соорудить цветную консоль.**

LINUX FORMAT Verdict

Mutt

Версия: 1.4.2.2
Сайт: www.mutt.org
Цена: Бесплатно под GPL

» *Трудный для первого раза, но головокружительно гибко настраиваемый и мощный почтовый клиент.*

Рейтинг 7/10

Evolution

Налицо все элементы персонального органайзера.

Это флагманский проект Gnome – интегрированные почтовый клиент, адресная книга и календарь. При запуске *Evolution* начинает выполняться Setup Assistant, предназначенный для установки в программе первоначальных учетных записей. *Evolution* поддерживает широкий диапазон протоколов, включая MAP, POP, Hula и Novell Groupwise, а также несколько опций локальной доставки *mbox/maildir*. Собственно говоря, опциями мастер перегружен, и это озадачивает, поскольку широкому большинству пользователей они никогда не понадобятся. Было бы куда приятней иметь что-нибудь попроще.

Evolution имеет виртуальные папки, так что вы можете сохранять отдельные результаты поиска, которые затем будут представлены как виртуальная папка в списке.

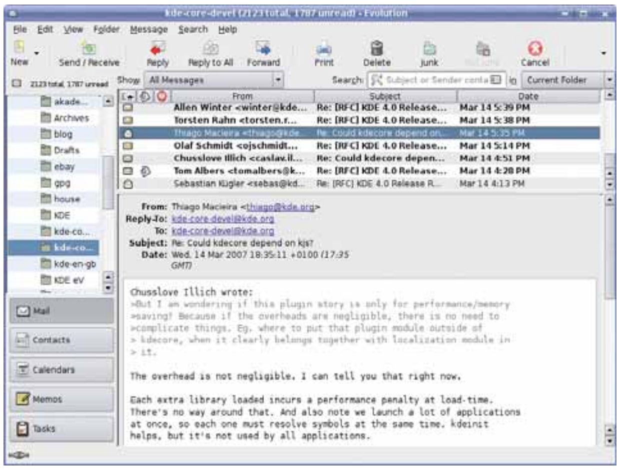
Чтобы не сидеть постоянно в Интернете в режиме IMAP, вы можете пометить папку как читаемую оффлайн. Тогда все её содержимое, до последнего письма, будет скачиваться на локальный компьютер.

Evolution имеет встроенную поддержку отправки и получения зашифрованных писем с использованием открытого клиента PGP, GnuPG, и стандарта S/MIME, а также детектор спама. Он работает хорошо и точно выявляет нежелательные сообщения.

Разработчики здорово упростили настройку *Evolution* с помощью функции под названием *Eplugin*, используемой для создания несложных модулей расширения – это изящный способ справиться с обычной проблемой необходимости марать руки исходным кодом на C. Система *Eplugin* намеренно разработана так, что вам хватит лишь минимального знания ядра приложения.

Производительность была потрясающей, *Evolution* потребовалось лишь 1,0 секунды для загрузки тестового почтового ящика. Поиск был произведен почти так же быстро, за 1,5 секунды – как по письмам, так и то темам.

> «Лишь секунда на загрузку тестового почтового ящика, и полторы секунды на поиск.»



› *Evolution* – не просто почтовый клиент: сюда включены календарь, адресная книга, заметки и многое другое.

LINUX FORMAT Вердикт

Evolution

Версия: 2.10.1
Сайт: www.gnome.org/projects/evolution
Цена: Бесплатно под GPL

» *Если вы ищете замену Microsoft Outlook – считайте, вы ее нашли.*

Рейтинг 7/10

Kontact (KMail)

Почтовый клиент от сообщества KDE.

Строго говоря, *Kontact* не есть почтовый клиент. Это обертка нескольких компонентов, включая календарь *KOrganizer*, адресную книгу *KAddressBook* и собственно почтовое приложение, *KMail*.

KMail имеет огромное число возможностей, но кое в чем ему недостает лоска, присущего другим приложениям. Например, опции настройки не сгруппированы логически, и иногда программа чересчур гонится за настраиваемостью: иллюстрация проблемы – вкладка с описанием Settings That Don't Fit Elsewhere (Настройки, которые никуда не подходят).

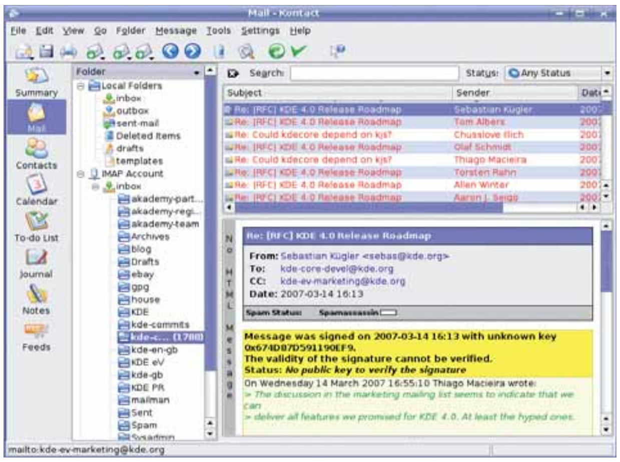
Как антивирус, так и фильтрацию спама предлагается настраивать мастером. Система фильтрации поддерживает несколько движков выявления спама – вы даже можете сами выбрать движок. Антивирусная защита нас разочаровала: она извещала об ошибке, но давала мало содействия в разрешении проблемы.

Средства конфиденциальности предусматривают отключение отображения HTML-писем: вместо них показывается голый HTML. Есть возможность включить отображение HTML-писем для всей папки, но это придется делать папка за папкой. К сожалению, в *KMail* отсутствует поддержка антифишинга. Зато имеется отличная поддержка шифрования с использованием *GnuPG* и S/MIME.

Недавно добавленная функция для написания HTML-писем охватывает всего несколько базовых возможностей, таких как цвет и начертание шрифта и опции форматирования. Есть поддержка чтения IMAP-почты оффлайн, но она прилагается ко всей учетной записи IMAP. Жаль, что нет возможности выбрать конкретные папки для оффлайн-чтения: либо все, либо ничего.

KMail'у потребовалось 12,6 секунды для загрузки тестового почтового ящика. Поиск по темам занял 12,6 секунды, а вот по содержимому писем – тягостные 115 секунд. Скорость загрузки существенно ухудшалась при попытках открывать в *KMail* большие почтовые ящики.

> «Фильтрация спама включает поддержку нескольких движков.»



› Подсветка зашифрованных и подписанных писем ясно показывает, кому можно доверять, а кому – нет.

LINUX FORMAT Вердикт

Kontact (KMail)

Версия: 1.9.6
Сайт: www.kontact.org
Цена: Бесплатно под GPL

» *Немного не хватает глянца и нескольких продвинутых антифишинговых функций, но на Kontact все еще следует обратить внимание.*

Рейтинг 7/10

Mozilla Thunderbird

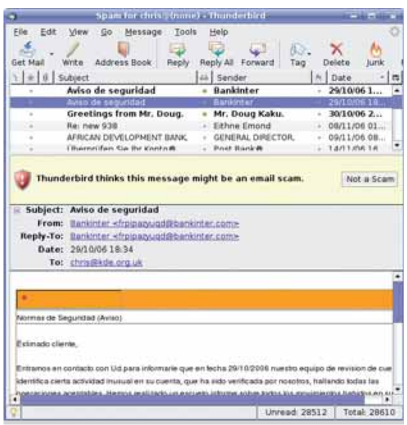
Флагманский почтовый клиент от Mozilla прекрасно сочетается с браузером Firefox.

Thunderbird был разработан Mozilla Foundation, некоммерческой организацией, также разрабатывающей широко известный web-браузер *Firefox*. Недавно вышедший *Thunderbird 2.0.0* содержит множество новых функций и исправлений, улучшивших предыдущую версию 1.5.

При запуске *Thunderbird* в первый раз вам дается возможность с помощью мастера установить новую почтовую или новостную учетную запись. Интерфейс *Thunderbird*'а по умолчанию представляет собой классический список папок слева, список писем вверху справа и содержимого письма внизу справа.

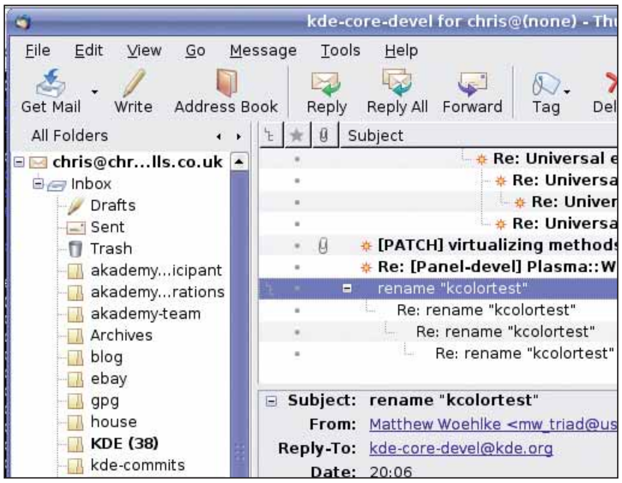
Ясно, что на разработку *Thunderbird* ушло немало мыслительного ресурса: здесь хорошо продуманный интерфейс пользователя и фантастические возможности в плане безопасности и конфиденциальности. По умолчанию, внешние рисунки в HTML-письмах не загружаются, это предотвращает обнаружение вашего почтового адреса третьими лицами. Злоумышленник, например, может внедрять рисунок с уникальным URL, связанным с вашим почтовым адресом, и если ваш почтовый клиент загрузит этот рисунок, атакующий получает подтверждение, что адрес существует и используется.

Thunderbird имеет встроенную систему определения спама по Байесу, и письма автоматически помечаются как спам и не-спам при скачивании. Если движок определения спама работает неправильно, есть возможность обучить его самому, помечая спам вручную. В предыдущих версиях *Thunderbird* с этой функцией была большая проблема, так как классификация, даже после обучения в течение нескольких недель, выглядела практически случайной; и, похоже, это существенно улучшено в *Thunderbird 2.0.0*. Без обучения она правильно определяла всевозможный спам и, что важнее, не объявляла ложных тревог, считая не-спам спамом.



› Щелчок на сомнительном URL выдает разумное предупреждение, что пойдя по этой ссылке, можно угодить на фишинговый сайт.

Thunderbird также имеет отличную функцию антифишинга, выдающую предупреждение '*Thunderbird* thinks this message might be an email scam' (*Thunderbird* думает, что это сообщение может быть обманным), при этом предлагая обучить движок, если письмо было опознано как обманное по ошибке.



› Пометив отдельные папки как 'Favorites' (Избранное), вы сможете переключаться между отображением всех папок или только избранных.

Ваша почта, сэр

Thunderbird автоматически проверяет новую почту через настраиваемые интервалы времени, и совершенно ненавязчивые всплывающие сообщения отображают в углу экрана тему письма и имя отправителя – это полезно, когда вы ждете важное письмо, но не хотите ежеминутно проверять почтовый ящик вручную.

Наряду с обычными POP3 и IMAP, *Thunderbird* также способен соединяться с Gmail, а также управлять локальными учетными записями почтового спула Unix. При работе с почтовыми учетными записями IMAP предлагается работа оффлайн, и вы можете выбирать, какие почтовые папки следует полностью скачать перед тем, как разъединиться с сетью.

Нам понравились кнопки Back и Forward (Назад и Вперед) на панели инструментов. Они работают аналогично кнопкам web-браузера – нажатие Back переносит вас на последнюю папку или письмо, которое вы просматривали, а нажатие Forward работает в обратную сторону.

У *Thunderbird* отличные инструменты поиска, с ними легко найти нужное письмо. Первый из них – это панель, возникающая внизу окна письма, позволяющая перемещаться вперед и назад при поиске по почте. Также есть панель поиска вверху списка писем, чтобы быстро отыскивать по отправителю или теме письмо в текущей папке. Наконец, *Thunderbird* может сохранять результаты поиска в папку, и если вам захочется провести поиск еще раз, все, что потребуется сделать – это перейти в папку и щелкнуть на имя соответствующего поиска.

Thunderbird использует гибкую модульную архитектуру, позволяющую создавать как дополнительные темы оформления, так и расширения (extensions), улучшающие функциональность. Расширения доступны в виде XPI-файлов. Они легко могут быть импортированы в *Thunderbird*.

Оплошность с шифрованием

Нас огорчило, что стандартно *Thunderbird* не имеет встроенной поддержки для получения и отправки PGP-подписанных или зашифрованных писем, но, к счастью, наличие расширений спасает положение, поскольку одно из них, под названием Enigmail, способно закрыть эту дыру.

Хотя это не совсем связано с электронной почтой, нам хотелось бы упомянуть, что *Thunderbird* также включает читалку RSS. Хотя у вас, наверное, уже есть любимый RSS-ридер для Linux, нам было приятно увидеть, что любимые новости могут быть прочитаны как письма (на стр. 25 показано, какие другие клиенты также имеют поддержку RSS).

> «Thunderbird обладает гибкой архитектурой модулей расширения.»

Производительность *Thunderbird* была хорошей, хотя и не выдающейся. Тестовый почтовый ящик открылся за 1,6 секунды, поиск по темам писем занял 2,0 секунды. А вот поиск по содержимому писем был вовсе не впечатляющ, на него ушло 58 тягостных секунд ожидания.

LINUX FORMAT Вердикт

Mozilla Thunderbird

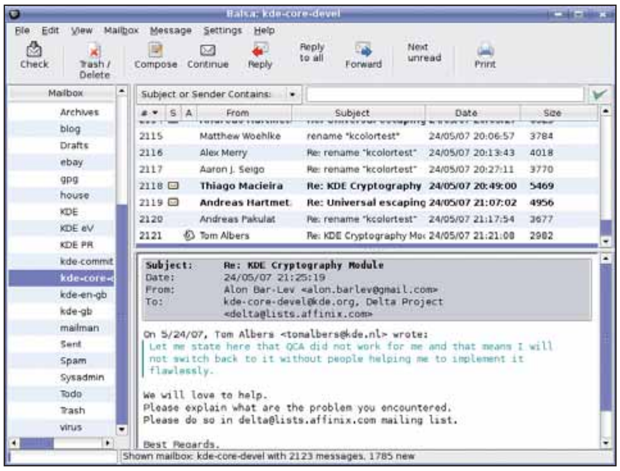
Версия: 2.0.0
Сайт: www.mozilla.org
Цена: Бесплатно под лицензией MPL

» Весьма приятное приложение – но хорошо бы ему обзавестись некоторыми групповыми функциями и календарем.

Рейтинг 9/10

Balsa

Простой, легковесный – не это ли ваш идеальный почтовый клиент?



› Наряду с использованием встроенных фильтров электронной почты, вы также можете использовать *Procmail* в связке с *Balsa*.

Balsa, будучи *GTK*-приложением, является частью рабочего стола Gnome. Но это не приложение все-в-одном, наподобие *Evolution* – оно разрабатывалось с упором на электронную почту, но при этом хорошо настаивается и предлагает все основные возможности почтового клиента.

Была сделана попытка улучшить безопасность, так как внешние рисунки, похоже, не загружаются в HTML по умолчанию. К сожалению, нам вообще не удалось найти способ их увидеть – не всплывает сообщения, как в других программах, с предложением отобразить их.

Имеется адресная книга, к которой можно обратиться через меню Файл, но она не оправдала ожиданий, поскольку выбор этого пункта приводил к сообщению об ошибке 'error opening address book "Address book": Cannot read from address book.' (ошибка при открытии адресной книги "Address book": Невозможно считать из адресной книги.) Оно возникало как на пакете Ubuntu версии 2.3.13, так и на собранной нами версии 2.3.15. Потратив некоторое время на поиск помощи, мы не сумели разрешить эту досадную проблему.

Нас озадачили некоторые аспекты пользовательского интерфейса *Balsa*. Примером может служить кнопка на панели инструментов с надписью Trash/Delete (Корзина/Удалить), тогда как подсказка говорит 'Move the current message to trash' (Переместить текущее письмо в корзину). Использование синонимичных выражений и несовпадение с формулировкой подсказки слегка сбивает с толку.

Поддерживаются фильтры электронной почты, позволяющие использовать регулярные выражения и даже указывать дату письма в некотором интервале времени. Быстрый поиск внутри папок конкретного отправителя очень прост, благодаря окну поиска над списком писем.

Производительность была одной из худших. Открытие тестового почтового ящика заняло 28 секунд, поиск по темам писем – 24,1 секунды. Поиск по содержимому писем вообще велся со скоростью черепахи, он занял около 176 секунд.

LINUX FORMAT Вердикт

Balsa

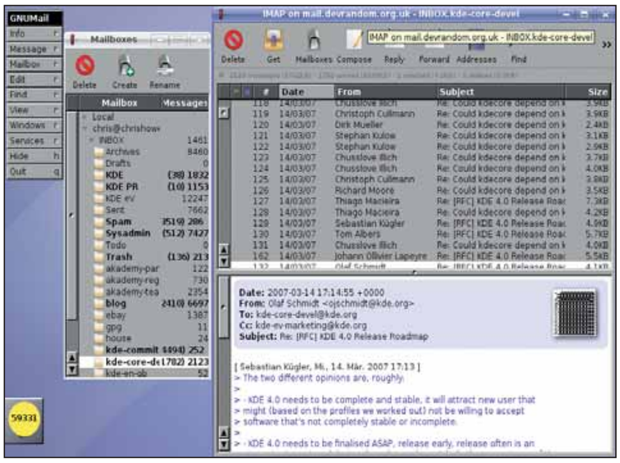
Версия: 2.3.15
Сайт: http://balsa.gnome.org
Цена: Бесплатно под GPL

» *Производительность у Balsa низкая, а набор возможностей – средненький.*

Рейтинг 5/10

GNUMail

Четырехлетний альтернативный пакет.



› К сожалению, невозможно писать и читать HTML-письма – отображается сырой HTML-код.

Запущенный в 2003 г., *GNUMail* является необычным клиентом электронной почты с уникальным пользовательским интерфейсом. Он разрабатывался как клон приложения *Mail.app*, поставлявшегося с операционной системой NeXT. *GNUMail* запускается на Linux и прочих Unix'ах, использующих среду разработки GNUStep.

Первый запуск *GNUMail* открывает диалог Preferences (Установки) на вкладке Accounts (Учетные записи). Он не так прост в использовании, как мастер, но куда доступнее некоторых своих собратьев. Кое-какие аспекты пользовательского интерфейса покажутся вам непривычными. Во-первых, полосы прокрутки расположены слева, а не справа. Во-вторых, меню расположено по вертикали, а раскрывается по горизонтали. Иногда это удобно. Меню обычно сжимается при выборе другого из пунктов сверху. Однако это можно предотвратить при раскрытии меню верхнего уровня путем перетаскивания его в другую часть экрана, используя заголовочную панель, затем отменить с помощью кнопки X, возникающей в панели меню.

Основной интерфейс *GNUMail* содержит два окна: окно меню и главное окно, где отображается список писем и показывается их содержимое. В другом окне можно выбирать другие почтовые ящики.

Установка фильтров писем проста благодаря различным пунктам меню, например Make Filter From Sender (Создать фильтр по отправителю). Откроется диалог построения фильтра, при этом поле Sender (Отправитель) уже будет заполнено нужной информацией.

Стабильность была ниже, чем полагается – попытка открыть папку, содержащую 30 000 писем учетной записи IMAP, привела к системной ошибке 'virtual memory exhausted' (исчерпана виртуальная память). *GNUMail* также не сумел завершить наш тест производительности, просто слетев при попытке скачать 60 000 писем, используя POP.

> «Полосы прокрутки слева, а не справа; меню расположено вертикально.»

LINUX FORMAT Вердикт

GNUMail

Версия: 1.2.0pre2
Сайт: www.collaboration-world.com/
Цена: Бесплатно под GPL

» *GGNUMail необычен, но пригоден к использованию, хотя требует доработки по части стабильности.*

Рейтинг 3/10

Почтовые клиенты

Вердикт

Thunderbird 9/10

Нам пришлось поломать голову, выбирая почтовых клиентов для этого **Сравнения** – можно было бы рассмотреть по крайней мере штук 15, без особого труда написав несколько страниц про каждый.

Выбранные же претенденты делятся на 2 четко разграниченные группы: чисто почтовые клиенты – такие, как *Mutt*, *GNUMail*, *Balsa*, *Thunderbird* и *Opera Mail*; и те, что включают возможности персональных органайзеров, например, календари и адресные книги – это *Kontact* и *Evolution*. Явный победитель выделялся на фоне всех претендентов почти по любому критерию. Это был *Mozilla Thunderbird*. Благодаря своей кросс-платформенности *Thunderbird* сейчас является одним из наиболее эффективных почтовых клиентов. Мы бы, однако, поторопили разработчиков включить в приложение возможности календаря. Отсутствие календаря и функций планирования и рассылки информации о встречах, скорее всего, не подпустит *Thunderbird* к корпоративным рабочим местам [необходимая функциональность разрабатывается в рамках проекта Sunbird. Одноименное приложение может как интегрироваться с *Thunderbird*, так и работать отдельно, но пока находится на ранних этапах развития, – прим. ред.].

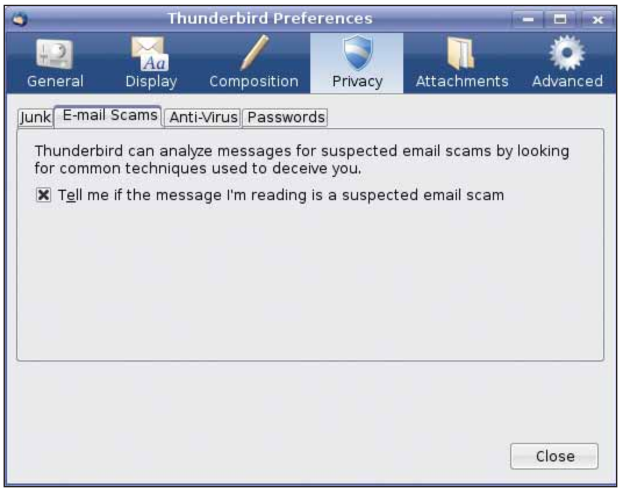
Несмотря на это, являясь внештатным семейным IT-специалистом, вы можете установить *Thunderbird* членам вашей семьи и радоваться тому, что они хорошо защищены от спама и фишинга.

Двумя другими выделяющимися клиентами были *Kontact* и *Evolution*. Оба способны эффективно заменить *Microsoft Outlook* – календарь, встречи и синхронизация адресной книги и электронной почты с другими устройствами – это просто Святой Грааль для настольного ПК с открытым ПО, ибо, как мы отметили, функции эти чрезвычайно нужны для бизнеса.

Kontact немного недошлифован, но, потратив некоторое время на его настройку, вы увидите, что это весьма мощный почтовый клиент. *Evolution* хорошо работает «прямо из коробки», но, похоже, некоторым областям было уделено мало внимания по сравнению с другими: например, мастер установки чрезмерно усложнен. LXF

Обратная связь

Мы проглядели ваш любимый клиент, или незаслуженно возвысили другой? Вам достаточно, что электронная почта помогает держать связь с семьей и сообществом, или же вы хотите увидеть больше возможностей для бизнеса? Присылайте ваше мнение с вашего любимого нового почтового клиента на letters@linuxformat.ru.



› Безопасность и конфиденциальность *Thunderbird* впечатляет.

> «Мы бы поторопили разработчиков Thunderbird добавить календарь.»

Таблица функциональных возможностей

	Balsa	Evolution	GNUMail	Kontact	Mutt	Opera Mail	Thunderbird
Лицензия	GPL	GPL	GPL	GPL	GPL	Проприетарная	MPL
POP	✓	✓	✓	✓	✓	✓	✓
IMAP	✓	✓	✓	✓	✓	✓	✓
Другие протоколы	Нет	Hula, Groupwise	Нет	Kolab, локальный	Локальный	Нет	Gmail
Антиспам	☒	✓	☒	✓	☒	✓	✓
Антифишинг	☒	☒	☒	☒	☒	✓	✓
Антивирус	☒	☒	☒	✓	☒	☒	☒
Адресная книга	✓	✓	✓	✓	Внешняя	✓	✓
Календарь	☒	✓	☒	✓	☒	☒	☒
HTML-письма	Только просмотр	Оба	Нет	Оба	Нет	Только просмотр	Оба
Фильтр	✓	✓	✓	✓	Внешняя	✓	✓
Модули расширения	☒	✓	✓ (Bundles)	✓	Макросы	☒	✓
Подпись/ шифрование	Оба	Оба	Оба	Оба	Оба	Нет	Оба, через модуль расширения
IMAP оффлайн	☒	✓	☒	✓	☒	☒	✓
RSS	☒	☒	☒	✓	☒	✓	✓
Проверка орфографии	✓	✓	✓	✓	В зависимости от редактора	✓	✓
Шаблоны	☒	✓	☒	✓	✓	✓	✓
Сохранение поиска	☒	✓	☒	✓	☒	✓	✓

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux.



ЛАДИСЛАВ БОДНАР
основатель, начальник, редактор и сотрудник DistroWatch.com.

Ветер перемен

«Я бы сказал, что для потребительского рынка продукт Windows, вероятно, пока остается более правильным выбором.» Эту мысль озвучил директор Red Hat Мэтью Зулик [Matthew Szulik] в ноябре 2003 г., несколько озадачив сообщество Linux. Крупнейший в мире поставщик Linux постоянно уверял, что Fedora – продукт, предназначенный скорее для отдельных энтузиастов-хакеров, чем для среднего пользователя, и не скрывал, что его приоритет – корпоративные серверы, а не настольные ПК.

Но все на свете меняется. Омоложенный проект Fedora, видимо, учел потрясающий успех, достигнутый за последнюю пару лет Ubuntu, и больше не считает настольную систему Linux исключительно достоянием хакеров. Новая команда художников превратила рабочий стол Fedora в один из самых элегантных, и достойна всяческой похвалы. Инициатива «Ноутбук – каждому ребенку» (One Laptop Per Child, OLPC), вместе с недавно объявленным Глобальным ПК (Red Hat Global Desktop) для развивающихся стран, тоже являются признаком, что Red Hat наконец пришла к выводу: настольная система – сектор рынка, за который стоит побороться.

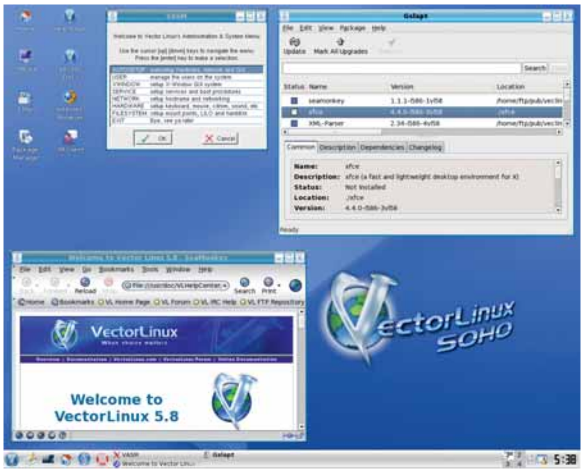
Добьется ли Fedora успеха? Это отнюдь не просто. Например, инсталлятор Fedora до сих пор не устанавливает автоматически двойную загрузку – а ведь это существенное удобство для технически не сильно грамотных пользователей, и главные конкуренты Red Hat предоставляют его уже несколько лет. Но все равно, отрадно видеть, что проект движется вперед. Такое ощущение, что от сообщества разработчиков Fedora повеяло свежим ветром перемен – ветром, который наконец принесет дистрибутив туда, где по праву его место: на столах миллионов пользователей.

ladislav.bodnar@futurenet.co.uk

Ускоримся!

VectorLinux 5.8 SOHO ОС на базе Slackware: приятнее, удобнее, да еще и с лучшей производительностью.

Не так уж много дистрибутивов Linux, существовавших на рубеже веков, все еще на коне и сегодня. VectorLinux – один из таких. Начатый Робертом Лэйнджем [Robert Lange] в конце 1999, проект канадского сообщества на базе Slackware регулярно выпускает стабильные версии и редко отклоняется от своей оптимистической цели: переделать стабильный, но чересчур технократический Slackware в более удобоваримый настольный дистрибутив. Изначально задуманный для старых компьютеров с ограниченными аппаратными ресурсами, проект также выпустил так называемую SOHO-версию, полноценную ОС с рабочим столом KDE и офисным пакетом.



› **VectorLinux SOHO – улучшенное издание проекта, с последним KDE, *OpenOffice.org* и *Mozilla SeaMonkey*.**

Если вы прошлый раз загружали VectorLinux несколько лет назад, вас ждет пара сюрпризов. Во-первых, VectorLinux за последние годы стал визуально привлекательнее, в основном благодаря своей команде оформителей. Это уже не просто пересборка Slackware с добавлением пары собственных утилит; это красивый, лощеный дистрибутив, достойный конкурент самых лучших на рынке.

Во-вторых, сделан акцент на удобстве использования. «Мы чувствовали, что пора предоставить нашим пользователям лучшее качество прямо из коробки, и в сегодняшнем мире это должно быть качественное видео», сказал Роберт Лэйндж. В результате, последняя версия VectorLinux обладает уникальными особенностями: улучшенный установщик, не только корректно определяющий видеокарту, но также при необходимости предлагающий во время установки собрать проприетарный модуль ядра от NVidia.

> «Красивый, элегантный дистрибутив; достойный конкурент самых лучших.»

Новая направленность на графику и мультимедиа также ясна по стандартному набору приложений VectorLinux. SOHO-издание проекта – один из немногих дистрибутивов, предлагающий *Xara Xtreme*, графический редактор для Unix, недавно открывший свой код. Включены также несколько приложений просмотра изображений (*KView*, *KuickShow*) и программы для воспроизведения видео (*VLC*, *MPlayer*). Поскольку штаб-квартира проекта расположена к северу от Великих Озер, то его не волнует патентное законодательство США на ПО и обход шифрования, и он преспокойно предлагает прямо из коробки поддержку множества патентованных медиа-форматов и шифрованных DVD.

Большая часть оригинального дистрибутива осталась неизменной. Инструмент администрирования VASM не столь подробен и не так прекрасно выглядит, как *DrakXTools* в Mandriva или Yast в OpenSUSE, но реально помогает настроить параметры системы. Менеджер пакетов *Gslapt* ни на бит не хуже *Synaptic* от Debian, и обеспечивает разрешение зависимостей и доступ к дополнительным пактам VectorLinux.

Текстовый установщик VectorLinux, вероятно, не дает ему стать действительно дружественной к пользователю альтернативой Slackware. Но, установив продукт на жесткий диск, вы найдете не много отличий от других настольных дистрибутивов на рынке – он быстр, имеет отличный выбор приложений, включает различные графические инструменты администрирования и предоставляет прекрасную онлайн-документацию.

http://vectorlinux.com

По линии студий

Ubuntu Studio 7.04 Новая версия Ubuntu, специально для работы со звуком.

Вам не казалось, что каждый день, проснувшись поутру, вы обнаруживаете где-то в Интернете новый вариант Ubuntu? В один прекрасный майский день это и вправду могло произойти: проект под названием Ubuntu Studio объявил о первом релизе своего дистрибутива для творческих личностей. Имея притягательное имя и обещая предоставить большой набор мультимедийного ПО, проект был мигом сметен из всемирной паутины тысячами алчущих пользователей, которые так и рвались на его раскаленный от посещений сайт.

До мастерской Ubuntu Studio было буквально не достучаться, но его система зеркал работала хорошо, и многие сумели загрузить установочный DVD размером 870 МБ. Немного печально, что Ubuntu Studio 7.04 – не LiveCD, а установочный образ, использующий старый текстовый установщик Debian. Тем не менее, после установки он превращается в полноценный дистрибутив – хотя и наследие Ubuntu, но с яркой темой рабочего стола, длинным списком аудио- и видеоприложений и специальным ядром с малым временем отклика, обеспечивающим суперпроизводительность при больших нагрузках.



› В Ubuntu Studio есть сэмплеры, запись звука, секвенсоры, звуковые редакторы, синтезаторы и много чего еще.

Стоит ли опробовать этого отпрыска Ubuntu? Как и всегда, это зависит от ваших видов на его применение, но согласимся, что если профессионалы видео, вероятно, сочтут, что приложения с открытым кодом уступают в сравнении с проприетарными аналогами (за исключением, естественно, стоимости), то музыканты откроют для себя в Ubuntu Studio огромный набор классных инструментов, которые помогут им превратить свои искания в прекрасные и нежные мелодии.

www.ubuntustudio.org

Краса Бразилии

Famelix 2.0 Красота – особенность этого южноамериканского дистрибутива.

Где бы ни проводился конкурс на красивейший дизайн рабочего стола Linux, все главные призы несомненно увезли бы разработчики одной страны: Бразилии. Популярный Kurumin Linux был первым дистрибутивом, поднявшим планку эстетики окружения рабочего стола, но он не остался одинок – следом пошли другие бразильские проекты, включая Kalango и Dreamlinux. Теперь к ним присоединился еще один мачо из крупнейшей страны Латинской Америки: Famelix.

Изначально целью команды Famelix было разработать дистрибутив, как можно более сходный с рабочей средой Windows. Его серия 1.x была разработана с полной имитацией вида и поведения Windows XP, но последний релиз, версия 2.0, пошел дальше: он не только клонировал рабочий стол Windows Vista, но также предоставил несколько улучшений. Уместно названный в рабочей версии Hasta la Vista («до свидания» – *исп.*), Famelix 2.0 предлагает поразительно красивый рабочий стол KDE, дополненный собственным K-меню, цветастым набором 3D-иконок, завораживающим фоновым рисунком и другими дополнениями рабочего стола, благодаря *SuperKaramba*.



› Это чарующее окружение было разработано в FAMEG, университете г. Гуарамирим, Южная Бразилия.

Famelix основан на старом Debian 3.1 Sarge, но поставляется с обновленными приложениями, включая последний KDE, а также *Firefox 2* и *OpenOffice.org 2*. В данный момент дистрибутив поддерживает только португальский и испанский языки, хотя есть признаки, что поддержка английского будет добавлена в одном из следующих релизов.

Famelix 2.0, работающий в качестве LiveCD, доступен для свободной загрузки с зеркал проекта. LXF

www.famelix.com.br

Пакеты на осмотре

Насколько новы приложения в вашем дистрибутиве? Приведенная ниже таблица показывает список основных компонентов всех наиболее известных на рынке дистрибутивов Linux. Пожалуйста, учтите, что OpenSUSE не выпускала нового релиза с декабря 2006 г. и выпуск следующей версии 10.3 планируется только в октябре 2007 г. Список пакетов Sabayon Linux соответствует его второму бета-релизу, но похоже, что финальная версия, которая должна уже выйти к тому моменту, когда к вам попадет журнал, будет иметь пакеты новее всех.

Дистрибутив	Ядро	Gnome	KDE	Firefox	OOo	Glibc	GCC
CentOS 5.0	2.6.18	2.16.0	3.5.4	1.5.0.10	2.0.4	2.5	4.1.1
Debian GNU/Linux 4.0	2.6.18	2.14.3	3.5.5	2.0.0.3	2.0.4	2.3.6	4.1.2
Fedora 7	2.6.21	2.18.1	3.5.6	2.0.0.3	2.2.0	2.5	4.1.2
Gentoo Linux 2007.0	2.6.19	2.16.2	3.5.5	2.0.0.3	2.1.0	2.5	4.1.1
Mandriva 2007.1	2.6.17	2.18.0	3.5.6	2.0.0.3	2.1.0	2.4	4.1.2
OpenSUSE 10.2	2.6.18	2.16.2	3.5.5	2.0	2.0.4	2.5	4.1.3
PCLinuxOS 2007	2.6.18	N/A	3.5.6	2.0.0.3	2.2.0	2.4	4.1.1
Sabayon Linux 3.4	2.6.21	2.18.1	3.5.6	2.0.0.3	2.2.0	2.5	4.1.1
SimplyMepis 6.5	2.6.15	N/A	3.5.3	2.0.0.1	2.0.2	2.3.6	4.0.3
Ubuntu 7.04	2.6.20	2.18.1	3.5.6	2.0.0.3	2.2.0	2.5	4.1.2

Хит-парад

10 самых посещаемых страниц на DistroWatch.com с 3 мая по 1 июня 2007 г. (среднее число визитов в день)

	Дистрибутив	Число посещений	
1	PCLinux OS	2,849	↑
2	Ubuntu	2,262	↓
3	Fedora	1,466	↑
4	SUSE	1,447	↓
5	Sabayon Linux	1,033	↑
6	Debian GNU/Linux	1,022	‹›
7	Mint	1,016	↑
8	Ubuntu Studio	764	–
9	Gentoo	711	–
10	SimplyMepis	710	↓

» **DistroWatch.com** оценивает популярность дистрибутивов, основываясь на числе посещений страницы каждого дистрибутива. Хотя это не дает представления о действительном числе установок, но показывает, какие дистрибутивы были более востребованы за определенный промежуток времени.

Клиент



David Blankenhorn

всегда прав



ДЖИМ О ЦЕНТРАХ ТЕЛЕКОММУНИКАЦИИ САН-ПАУЛУ

«Существуют люди, у которых есть адреса электронной почты, но настоящего адреса нет — они бездомные.»

Джим МакКиллан и **Рон Колсернян** основали проект Linux Terminal Server Project (LTSP) восемь лет назад, и с тех пор он стал трамплином для всех желающих дешево внедрить у себя терминальные технологии. *Linux Format* встретился с ними.

На заре распространения компьютеров всем правил центральный сервер, а пользователи получали доступ через «безголовый» терминал, просто передававший данные ASCII с клавиатуры на сервер и обратно для их отображения в виде зеленого текста на черном экране. Появление настольных ПК в корне изменило этот подход. Но в последнее время новая технология терминалов возродила интерес к концепции центрального сервера и к преимуществам, которые он предлагает в области снижения затрат и управляемости. Поэтому новое значение приобрел проект Linux Terminal Server Project (LTSP), созданный Джимом МакКилланом [Jim McQuillan] и Роном Колсерняном [Ron Colcernian] в 1999 г., чтобы помочь людям использовать Linux и дешевое оборудование для реализации данной модели сетевого взаимодействия. **Ник Вейч** помчался к ним в Штаты, чтобы узнать, чего добился проект почти за десять лет своего существования.

Linux Format: **По-моему, мы познакомились с вами в 2001 году на выставке LinuxWorld в Нью-Йорке, как раз когда начинался LTSP.**

ДжМкК: Ну, начали-то мы летом 1999, представив проект общественности. В 2000 году на LinuxWorld у нас был стенд пополам с Etherboot, а в 2001 мы, наверное, впервые обзавелись собственным стендом. Так что мы тогда и вправду еще находились в младенчестве.

LXF: **Полагаю, за прошедшие годы многое изменилось?**

ДжМкК: Технология претерпела некое развитие, а вот основные идеи не особенно изменились. Я думаю, наш продукт стал лучше, потому что он повзрослел. Мы добавили такие функции, как поддержка локальных устройств – вы подключаете USB-брелок, и на рабочем столе появляется иконка. Вы скажете – это не бог весть что, все так делают, но не забудьте, что в нашем случае сессия запущена на сервере, а пользователь сидит за тонким клиентом. Если вы подключаете к клиенту устройство USB, сессия на сервере должна знать об этом. Мы все сделали, как надо, чтобы все хорошо работало; это очень элегантное, прочное решение, без всяких хакерских штучек, которыми мы увлекались раньше. Оно действительно отлично сделано.

Еще нам помогало то, что приложения под Linux становились очень хорошими. Когда мы только начинали, не было текстовых редакторов, web-браузером был Netscape 4.x, вполне приличный, а клиента электронной почты и всего прочего просто не существовало. А теперь у нас есть офисный пакет *OpenOffice.org*, у нас есть *Firefox* и *Thunderbird*, *Evolution* и всяческие прекрасные, богатые функциями приложения. Все они росли вместе с нами, облегчая нашу работу. Пустившись устанавливать LTSP в 2001 году, когда мы познакомились, вы изрядно попотели бы. Надо было выполнить множество шагов по инструкции. Но LTSP сильно вырос. Сейчас его установка очень проста. Вы скачиваете небольшой пакет *LTSP_utils*, »

» содержащий инсталлятор, запускаете его, и он устанавливает и настраивает систему для вас – это очень легко проделать. До настоящего момента LTSP занимался именно этим.

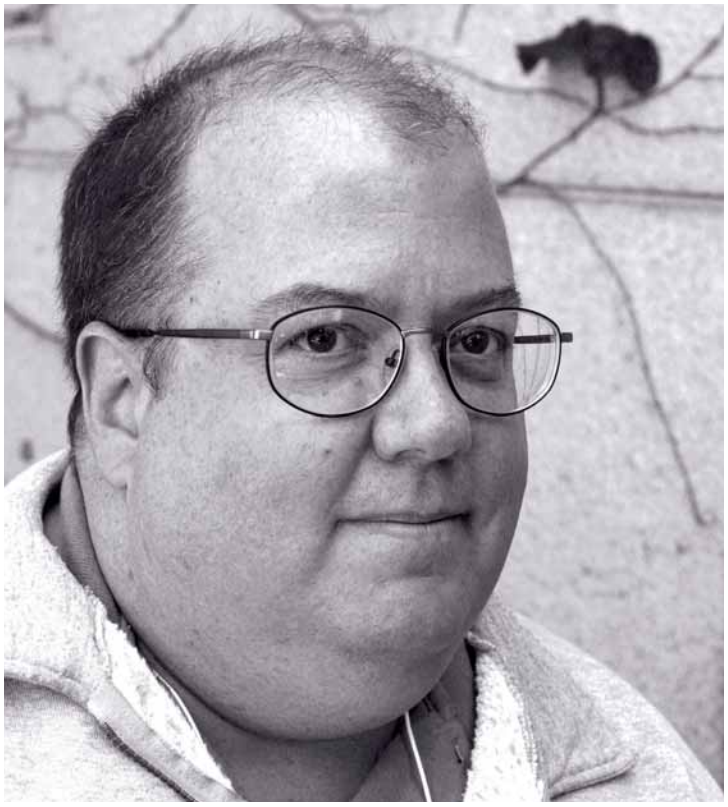
Года полтора назад мы связались с Ubuntu, и сейчас LTSP интегрирован в дистрибутив Ubuntu. А сейчас наша забота – собрать вместе всех основных игроков: мы добились того, что Red Hat, Debian, Ubuntu и некоторые другие сядут вместе и обсудят, как будет тонкий клиент работать в дистрибутивах Linux, чтобы все дистрибутивы смогли включить LTSP. Моя цель – встроить в дистрибутив функциональность тонкого клиента. Вы устанавливаете свой дистрибутив, активируете ее, подключаете тонкий клиент – и вперед. Никаких лишних движений типа посещения нашего сайта и скачивания кода. Он уже должен быть там: *Apache* и *Samba* уже есть – и LTSP тоже будет.

LXF: Полагаю, что на данный момент с одними дистрибутивами это делать легче, чем с другими?

ДжМкК: Да, иногда из-за философии, иногда – по техническим причинам, но пока все, с кем я разговаривал по этому поводу, с большим воодушевлением принимали эту идею. Ubuntu работал над ней полтора года, и Ubuntu и Debian постоянно обмениваются друг с другом функциями, разрабатываемыми в Perl, чтобы включить LTSP. Разработчики Ubuntu и разработчики Debian очень тесно сотрудничают. Так что это практически одна и та же реализация. Fedora, естественно, будет основан на RPM, а не на пакетах Deb; SUSE будет основан на RPM, хотя определенно не на тех, которые использует Red Hat. Цель у нас следующая: когда мы закончим работу, LTSP будет установлен на нашем сервере, а тонкий клиент будет загружать в случае Ubuntu – части Ubuntu, в случае Fedora – части Fedora. На LTSP.org нашей главной работой будет обеспечить людей тем, что позволит тонкому клиенту работать как я вам описал. Несколько лет мы предоставляли библиотеки, и ядро, и все остальное, а с этим отлично справляются дистрибутивы, вот пусть они и делают свое дело, а мы выходим из этого бизнеса и будем обеспечивать людей тем, чем нам полагается их обеспечивать.

LXF: Я собирался спросить, что изменилось в проекте LTSP за время его существования благодаря развитию дистрибутивов.

ДжМкК: По традиции, вплоть до настоящего момента, когда вы устанавливали LTSP, это был исключительно LTSP, и мы хотим покончить с этим. Я написал спецификацию по данному предмету; Марк Шаттлворт [Mark Shuttleworth] ее прочел, и на ней частично теперь базируется версия Ubuntu. В своей спецификации я писал:



«Взгляните: 95% того, что мы поставляем с LTSP, не имеет ничего общего с LTSP – просто оно должно быть, потому что LTSP получается вроде дистрибутива, который мы сажаем поверх основного дистрибутива. Так давайте мы переведем все это в основной дистрибутив и будем поставлять наши 5%. Лучше мы сконцентрируемся на таких важных вещах, как синхронизация с КПК, или прочная поддержка мультимедиа, или безопасность и тому подобное».

LXF: Какие, по-вашему, ближайшие задачи встанут перед вами после интеграции с дистрибутивом?

ДжМкК: Мне бы очень хотелось заняться следующей важной задачей – хорошей, солидной поддержкой мультимедиа; но сейчас мы не работаем над ней из-за всей этой возни с дистрибутивами. Надо, чтобы вы могли воспроизводить видео и аудио, и сигнал был бы должным образом синхронизирован, и чтобы при этом на сервере было 30 или 40 клиентов, воспроизводящих либо то же самое, либо что-то другое. Пусть люди делают все то же самое, что и в Linux, но уже как пользователи тонкого клиента – вот это и есть наша следующая задача.

LXF: Вас об этом просили, или это просто такая личная цель?

ДжМкК: Да, об этом просят, и этого ждут. Раньше нам приходилось привлекать внимание, чтобы сказать: «Нам нужна помощь с аудио в KDE» или «Нам тут нужна помощь кое с чем в Gnome». И я всегда испытывал такое ощущение, что вот они – такие крупные, большие проекты, а мы – мелкота. А теперь мы сидим за одним столом с этими парнями. Мы садимся и разговариваем с человеком из Phonon – это звуковая подсистема в KDE. Мы разговариваем с разработчиками *GStreamer*, это используется в Gnome. Мы разговариваем с другими разработчиками Gnome о том, что нам нужно. И все реагируют примерно так: «Здорово, нам это нужно, давайте сделаем это». Люди понимают, что LTSP – ценная возможность установить их приложения на множестве настольных компьютеров. И меня это воодушевляет.

LXF: А какова, по вашему мнению, причина возврата интереса к тонким клиентам?

ДжМкК: Люди устали от того труда, который приходится вкладывать в поддержание инсталляций по типу Windows. В самом деле, на каждом столе стоит компьютер, в каждом хранится множество данных – весьма много приходится поддерживать. С точки зрения администратора, надо следить, чтобы очень много всего работало, все время.

РК: И вы многим рискуете, потому что работники неизбежно будут хранить электронные документы, за создание которых вы им заплатили – десять долларов, несколько тысяч долларов – на носителях без резервных копий. А в LTSP у вас по умолчанию создается централизованная резервная копия всего рабочего пространства пользователя, и сама структура программы снимает весь этот риск. Я не знаю, сколько раз приходится слышать, как кто-то говорит: «Мне придется заново делать эту таблицу». Как владельца бизнеса, это просто бесит.

LXF: Да, потерянное время и потерянные деньги.

ДжМкК: Люди начинают осознавать, что во многих случаях тонкий клиент – вещь весьма удобная. Он не идеально подходит к каждой ситуации: для редактирования видео или еще какой-то крупной задачи он, возможно, и не годится, но подумайте, скольким людям нужен просто текстовый редактор, или редактор электронных таблиц, web-браузер, клиент электронной почты, а больше и ничего. Ну, может, еще какое-нибудь унаследованное приложение для передачи данных в базу. Для таких задач тонкие клиенты подходят идеально.

LXF: Я полагаю, именно для офисного пользователя, как это называется в Novell, и был разработан SLED.

ДжМкК: И тонкий клиент может стать отличным способом донести его до множества рабочих столов.

LXF: Как вы считаете, почему для настройки тонкого клиента выбирают именно Linux, а не что-либо другое?

ДжМкК: А не Windows? Многие просто уже устали от игры, которую навязывает Windows: постоянное обновление программ и оборудования. Рон намного лучше это объясняет, чем я! Это просто ад, через который они проходят. Хотите добавить что-нибудь, Рон?

РК: Людей мотивирует то, что они устали от Windows. Я сегодня говорил с одним парнем. Он сказал: «Вот я пользуюсь Windows и отлично зарабатываю на этом». А я сказал: «Может, и так. Но давай поговорим об одном-единственном дне, кото-

рый уходит на решение проблем Windows. Дне, который ты тратишь на загрузку и перезагрузку программ по восемь часов». И в конце этого дня ему платят, но, наверное, в отличие от вас, и от меня, и от Джима, он совершенно не обогатил свой мозг интеллектуально. Он не увеличил свою ценность для общества, он не увеличил свою самооценку. Я думаю, многие начинают над этим задумываться.

LXF: По-вашему, большинство просто сразу перейдет с Windows на настольном ПК на Linux в тонком клиенте, или они предпочтут делать это постепенно?

ДжМкК: Бывает по-разному, в зависимости от компании.

РК: Я думаю, ответ лежит в количестве развертываемых рабочих мест. Вот сегодня приходил один человек, у него центр обработки звонков, по-моему, мест на 75. И он понятия не имел об LTSP. Он решил (причин он не указал) перейти с Windows на Linux. И они установили толстые клиенты с полными дистрибутивами Linux на каждом столе. А если бы они имели возможность рассмотреть наш проект или познакомиться с ним, они бы сразу рванули на тонкие клиенты, потому что это намного эффективнее. Возможно, смысл в том, чтобы чаще выдавать информацию на сторону: если сделаешь это больше, скажем, четырех, пяти раз – какого-то магического числа – то технологию уже начинают замечать.



› Джим на каждом шоу пользуется случаем донести до масс благую весть о LTSP.

LXF: Вас никогда не огорчало, что очень многие воспользовались бы таким решением, как ваше, если бы они знали о нем?

ДжМкК: Мне всегда хотелось, чтобы у нас лучше получалось рассказывать о себе; больше людей могли бы о нас узнать, потому что, по-моему, чем больше людей этим пользуются, тем лучше. У меня есть несколько историй успеха, о которых я думаю с гордостью. Пусть не все об этом знают: это неважно. Я не лишусь из-за этого сна. Те, кто пользуется LTSP, довольны им.

LXF: И какая же у вас любимая история успеха?

ДжМкК: От которой мурашки по коже? Их несколько. У нас огромная работа в Сан-Паулу, в Бразилии. Шесть тысяч телекоммуникационных центров – это такие киберкафе, спонсируемые правительством – 6 000 киберкафе, 20 рабочих станций и сервер в каждом. И рабочие станции работают на LTSP. Это 120 000 терминалов по всему городу, а Сан-Паулу – огромный город, третий по величине в мире. Там есть люди, у которых есть адрес электронной почты, а настоящего адреса нет – они бездомные. Но они все-таки могут общаться со своими семьями, друзьями, с кем угодно. Они могут бродить по сети, находить контакт с правительством: подавать заявления на получение социальной помощи, и все такое. Этот пример действительно меня воодушевляет.

РОН О ПЕРЕХОДЕ НА LINUX

«Мотив перехода на Linux на тонких клиентах – люди устали от Windows.»

А вот что сделал Фонд Шаттлворта [Shuttleworth Foundation] в Южной Африке: они предоставили 240 000 компьютеров для школьников. Это не 240 000 рабочих станций, но они обслуживают множество рабочих станций в каждой школе, для 240 000 ребят, которые иначе никогда не получили бы доступа к компьютерам. И это очень важно: эти дети выйдут из школы с такими же навыками – а часто и с лучшими – чем такие же дети из Америки. Вот от таких примеров у меня мурашки по коже, они вдохновляют меня на движение вперед.

LXF: Если заглянуть в будущее, что бы вы хотели, чтобы произошло через десять лет?

ДжМкК: Через десять лет? Это долго. Проекту только семь лет. Давайте немного сократим срок: через три года я хочу видеть LTSP во всех дистрибутивах. Потом, я хочу сфокусироваться на трудных частях – на развлечениях. Чтобы люди просто… ну, этак естественно устанавливали Linux, подключали к нему несколько терминалов, и все готово. Вот чего я хочу, а через десять лет – это трудно себе представить.

LXF: Помимо потокового мультимедиа, о котором мы говорили раньше, какие еще технические проблемы, по-вашему, требуют решения?

ДжМкК: Большая проблема – это безопасность. Мы используем NFS, мы используем X, мы используем DHCP и TFTP, чтобы загрузить эти клиенты. И там большие проблемы с безопасностью. Это не проблема для школ или малых офисов, где всего штук 15 терминалов и нет посторонних глаз, но в финансовых учреждениях мы работать не можем, из-за используемых у нас протоколов. Так что мы рассматриваем такие вещи, как работа IPSec между терминалом и сервером, или можно ли запустить X через SSH для шифрования X-трафика. Это здорово обременяет сервер, но сервера в наши дни стали довольно крупными.

Именно в этой области мы и должны работать, как только разберемся с мультимедиа. По мере вовлечения все большего числа людей и дистрибутивов, увеличивается и оказываемая нам помощь в решении всех этих проблем.

LXF: Я как раз хотел спросить о этом: некоторые проекты так и не вырастают за пределы основной команды разработчиков, начавших проект; а вам многие присылают код?

ДжМкК: Я получаю множество заплаток, множество всего. У нас есть wiki, чтобы люди могли выкладывать туда то, что у них есть, у нас есть канал IRC для всех, кто захочет общаться с нами – мы делаем все наши разработки через IRC.

Мы действительно работаем открыто – никаких частных разработок: все в IRC или в списке рассылки. Туда заходят люди, одни помогают нам недолго, кто-то остается на большой срок. Сейчас мы около полутора лет работаем с Ubuntu, и есть целая команда разработчиков, связанных с этим дистрибутивом, которым интересно то, что мы делаем. И команда Fedora вливается в процесс, делая то же самое…

LXF: Я недавно разговаривал с парнями из Fedora, и они демонстрировали большое воодушевление…

ДжМкК: На хакфесте LTSP в сентябре к нашей работе собирается присоединиться Уоррен Тогами [Warren Togami]. Он создал Fedora. Он начал проект; Red Hat пригласил его на работу в качестве разработчика. Он собирается работать с нами; он хочет, чтобы LTSP был интегрирован в Fedora. Для него это важно, а значит, это становится важным и для других людей. К нам присоединяются масштабные личности, и это превосходно.

LXF: Как вы думаете, чем лучше всего смогли бы вам помочь обычные пользователи?

ДжМкК: Есть много разных способов помочь нам. Они могут писать код, могут помогать нам в решении некоторых проблем, над которыми мы бьемся. Если они умеют писать документацию, пусть помогут нам в разработке документации… Определенно могут помочь в переводе. У нас есть команда переводчиков, которой пригодились бы люди. Если они хотят нести информацию в массы, это вообще отлично – в смысле, они могут помочь в популяризации. Если они захотят посидеть в канале IRC пару часов в день, отвечая на вопросы посетителей – и это будет огромная помощь. Или просматривать списки рассылки и периодически отвечать на присланные вопросы, тоже отличный способ помочь. LXF

СПРАВОЧНИК: ЖЕЛЕЗО–2007

Стойте! Прежде чем пойти и потратить денежки на новую периферию для вашего PC, прочитайте это руководство Пола Хадсона и Майка Сондерса о том, что работает, а что нет.



Вам ненавистны люди, чьим первым дистрибутивом Linux был Ubuntu? Они говорят: «Почему все думают, что Linux – это сложно? У меня сразу все заработало!» Ладно, «ненавидеть» – слишком сильное слово, но многие из нас побывали в ситуации, когда радостно подключаешь какое-нибудь модное новенькое устройство под Linux с тем, чтобы получить… ничегошеньки. Ни даже звука. Windows 98, слетая при подключении чего-нибудь нового, хотя бы показывала: что-то произошло.

Много лет мы принимали как данность то, что Linux воротит нос от большинства предлагаемого оборудования, но новая порода дистрибутивов попыталась изменить это. Ubuntu, при всем своем популизме, действительно работает с первого раза у большинства людей. Однако проблема совместимости с оборудованием не решена до конца – недостаток поддержки от производителей обычно означает, что нужно дожидаться, пока какие-нибудь хакеры ядра не купят это устройство, потому что до этого драйверов нам не видать. Пытаясь переломить ситуацию, команда разработчиков драйверов для Linux недавно запустила схему свободной разработки: они берутся писать и поддерживать драйверы устройств любой компании, которая снабдит их оборудованием и спецификацией его работы.

Плоды этой деятельности еще не созрели, но вопрос остался: насколько хорошо работают устройства под Linux прямо сейчас? Для ответа на него мы протянули свои рабочие перчатки к тому оборудованию, что нашли в закромах магазинов по соседству с офисом *Linux Format* в издательстве Future Publishing. Было тут и старое, было и новое, попадалось слегка странное и с причудами, но все это можно купить в магазине или через Интернет: наша задача – помочь вам оценить, что будет работать и что не будет, при очередной закупке снаряжения.

Обозначения ★★★★★

★ .... Смеется в лицо совместимости с Linux

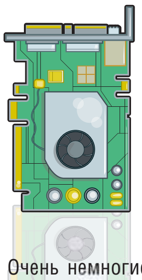
★★ .... Работает лишь малая часть функций; головная боль обеспечена

★★★ .... Работоспособно; требует эквилибристики с командной строкой

★★★★ .... Работает хорошо, с любовью

★★★★★ .... Все функции, поддерживаемые Windows или OS X

Внимание: Цены, приведенные в данной статье, были адаптированы для российского рынка и получены усреднением данных Price.Ru и Яндекс.Маркет. В случае, когда установить стоимость устройства в РФ не представлялось возможным, в графе «цена» указывалось «нет», а в скобках – приблизительная стоимость устройства за рубежом, вычисленная на основе данных Ebay и Froogle. Все цены были действительны на момент подготовки статьи.



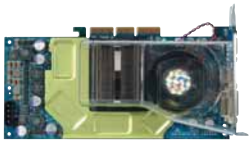
Видеокарты

Раньше ATI отставала от Nvidia в поддержке драйверов, но нам говорили, что это уже не так. Наш ответ: все по-прежнему плохо.

Очень немногие перешли с Windows на Linux ради игр. Точнее сказать, никто не перешел с Windows на Linux ради игр. Но это не значит, что люди захотят расстаться с любимыми Windows-играми, и если не прибегать к двойной загрузке, остается лишь желать, чтобы их графическая карта работала под Linux. Есть и другая группа людей – им нужна видеокарта для причудливых эффектов рабочего стола *Compiz*, но эта группа довольно мала, да и к тому же можно добиться выполнения большинства этих эффектов на скромной видеокарте Intel.

Для серьезной 3D-работы существуют только два изготовителя графических карт, которых стоит рассматривать: Nvidia and ATI. Мы взяли две старых карты (GeForce 5950 Ultra и Radeon 9800 Pro) и три новых карты (GeForce 8800, Radeon 1950 и Radeon 2900) и протестировали их на Ubuntu 7.04...

Nvidia GeForce 5950 Ultra



Эта видеокарта начиняла **Ультракомпьютер с Linux** в 2004 г. (см. LXF92) и служила нам хорошую службу в бесчисленных играх, в которые мы резались: *Postal 2*, *UT2004* и другие игры той эры или старше гладко работали при установках высокого разрешения, а *Doom 3*, *Quake 4* и другие более новые игры отлично справлялись на среднем качестве. Понятно, что если вам подавай разрешение выше, чем 1024x768, то придется уменьшить (или отключить) сглаживание и другие дополнительные возможности, однако при отказе от излишней детализации можно дойти и до 1600x1200. Поддержка Linux слегка отставала при покупке устройства, но оно безупречно работало как минимум последние два года и даже поддерживается установщиком бинарных драйверов Ubuntu.

Цена: нет ($60)

Рейтинг: ★★★★★

ATI Radeon 1950 XT



Поскольку это сравнительно свежая видеокарта ATI, мы подошли к делу с осторожностью. ATI никогда не имела хороших драйверов для Linux, но нам многие говорили, что времена изменились в лучшую сторону и драйверы ATI теперь не хуже Nvidia. Что ж, под Windows это действительно хороший исполнитель, легко достигающий производительности видеокарт Nvidia 7900. Но под Linux мы вскоре завязли в трясине ATI-драйвера: менеджер двоичных драйверов Ubuntu вызывал неоднократные проблемы, официальный ATI-драйвер отказался работать в 3D-режиме без объяснения причин, и в конце концов мы остались в режиме VESA.

После мучительного разбирательства с менеджером драйверов, переустановки двоичного драйвера и ручного редактирования файла конфигурации X, чтобы ATI-драйвер правильно работал с Composite, GLX и DRI, мы наконец получили работающую поддержку 3D. Возможно, что с более старыми видеокартами все проще, но 1950 – последняя видеокарта в линейке, и мы хлебнули с ней столько горя, что чуть было не сдались.

Цена: 5700 руб.

Рейтинг: ★★★

ATI Radeon 9800 Pro



Это – главная опора малобюджетных видеокарт, и ее можно найти по цене дешевле хорошей выпивки, если пошарить по магазинам. Хотя она и будет стараться изо всех сил, ее не рекомендуется использовать для всего, что вышло после *Unreal Tournament 2004*, и даже в этом случае придется следить за используемым разрешением. Что касается поддержки Linux, это один из немногих случаев, когда ATI поддерживается чисто: вставьте карту, используйте установщик двоичных драйверов – и вы в режиме 3D без лишних вопросов.

Цена: нет ($50 б/у)

Рейтинг: ★★★★★

ATI Radeon 2900 XT



Не заработала ни на самую малость, и потратив много часов, пытаясь заставить ее сделать хоть что-нибудь, мы пришли к выводу: непохоже, что она вообще заработает. Держитесь подальше!

Цена: 13 000 руб.

Рейтинг: ★

Nvidia GeForce 8800 GTS/GTX



GTS-версия этой карты – максимальная смесь цены и мощности: как GPU GForce 8 серии, она способна выполнить все, что вам хочется, на максимальном разрешении. Если вы хотите поиграть в новый *Unreal Tournament* на разрешении, большем 800x600, эту-то видеокарту вам и надо. Если у вас есть лишние деньги, версия GTX позволит вам играть на разрешении 1920x1200 без малейших пропусков. Что касается поддержки Linux, она там, где в свое время был 5950 – вам потребуется перезагрузиться в режиме VESA и установить двоичный драйвер Nvidia вручную. На старых дистрибутивах, в том числе и на Ubuntu 6.10, видимо, возможны проблемы с работой видеокарты GTX в режиме VESA, но это исправлено в 7.04, а у SUSE, кажется, проблем вообще не было.

Установка двоичного драйвера на Ubuntu 7.04 была непростой, поскольку ее следует производить, когда X не запущен. Почему-то завершение GDM и X для попадания в командную строку оставляло нас с пустым экраном, то же самое получалось по Ctrl+Alt+F1. Нашим решением было открыть терминал X и запустить **sudo rm /tmp/.X0-lock** – без этого файла установщик Nvidia не догадывается, что X запущен, и драйвер можно благополучно установить. Затем мы столкнулись с еще одной проблемой, из-за попытки использовать установщик бинарных драйверов Ubuntu. Оказалось, что драйвер, который Ubuntu услужливо устанавливает для вас, слишком стар, чтобы поддерживать видеокарты линейки 8800! Наше решение: не связывайтесь с этим установщиком.

Цена: GTS 10 000 руб., GTX 20 000 руб.

Рейтинг: ★★★★



Принтеры

Да, мы в курсе насчет исчезновения лесов и безбумажного офиса, но иногда просто необходимо нажать Печать!

Ничего не бесит нас больше, чем принтер, не работающий под Linux, потому что факты таковы: за последнее десятилетие ничего удивительного из мира принтеров не вышло, следовательно, сейчас-то все должно работать? Увы, ситуация не столь радужная. Фактически, если ваш принтер не HP или Epson, то шансы на успех резко падают. Но многие изготовители, традиционно открещивавшиеся от Linux, постепенно обзаводятся неофициальной поддержкой через такие проекты, как Gutenprint (бесплатно) и TurboPrint (платный); а значит, Canon'ы и Lexmark'и снова на повестке дня – по крайней мере на настоящий момент.

HP PhotoSmart 3310



Многофункциональные устройства, подобные этому, имеют плохую репутацию у Linux – в прошлом у вас могла заработать печать, или сканирование, или встроенный привод для чтения карт памяти. Но данное устройство от HP поразило поддержкой всего этого прямо из коробки – открытый инструментарий HP LIP немедленно распознал принтер, и SANE подхватил сканер с минимальной морокой. Качество печати не уступало Windows и OS X, без всякой специальной настройки, и можно успешно удаленно отслеживать уровень чернил. Одним словом, безупречно.

Цена: нет ($600)

Рейтинг: ★★★★★

Epson Stylus Photo R2400



Epson'овская линейка Stylus всегда оставляла позади продукцию HP, располагая фотораспечатками сверхвысокого разрешения и моделями для формата A3. Но, к сожалению, на этом принтере мы не смогли достичь того качества под Linux, какое было под OS X – текст выглядел так же, а вот супер-фотораспечатки из-под Mac выглядели ярче, более насыщенно и менее зернисто, чем из-под Ubuntu. Тем не менее, нам известно, что идет работа по улучшению драйвера через Gutenprint, значит, ситуация будет исправлена.

Цена: 27 000 руб.

Рейтинг: ★★★★

Списки совместимости оборудования

Независимо от вашего дистрибутива, отыщите Список совместимого оборудования (Hardware Compatibility Lists – HCL) где-нибудь на его сайте, это даст вам ряд указаний на известные проблемы с периферийными устройствами, чипсетами и другими частями архитектуры компьютера. Если ссылка сразу не нашлась, заглавная страница www.linuxdrivers.org содержит обширный список совместимых устройств для большинства дистрибутивов. Есть также дружественные форумы и многие другие ресурсы: www.linuxcompatible.org.

HP Deskjet 970



Это был один из немногих дешевых струйных принтеров, который не прекращал работать и не терял поддержку производителя через год после покупки. Его возраст и простота гарантировали совместимость, и в самом деле его можно заставить работать либо через Gutenprint, либо через официальный инструментарий HP-LIP – мы бы порекомендовали последнее, поскольку он полностью автоматический и имеет более высокое качество фотопечати. Имеются некоторые причуды, например, неспособность печатать черный текст при пустом цветном картридже, но мы обнаружили, что такая же проблема присутствует в OS X, так что ясно, что здесь дело не в Linux.

Цена: нет ($50 б/у)

Рейтинг: ★★★★★

Модель на фото не является именно той, которая тестировалась

Canon PIXMA iP1800



Мертвый по прибытию, мертвый после часов попыток различных Canon-драйверов, мертвый после вмешательства Gutenprint, и мертвый после бесчисленных молитв. Он мертв; Мария, Елена, Роман, Тимофей, Виктор. Canon имеет официальную поддержку Linux для некоторых принтеров (на www.canon-europe.com/Support/software/linux), но этого в списке не оказалось. Также он не поддерживается Gutenprint или TurboPrint, и если вы любитель данного устройства, считайте себя покинутым. Лучше поискать iP4200.

Цена: 1500 руб.

Рейтинг: ★

Lexmark X1290

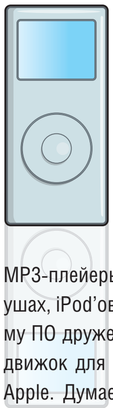


Этот принтер до того дешев, что так и подмывает плюнуть на поддержку Linux и купить его. Мы серьезно. Но так как мы здесь судим о совместимости с аппаратным обеспечением, мы покорно воткнули его и ждали, пока Linux даст нам знать, заработает устройство или нет. Сюрприза не вышло: оно не заработало. Lexmark имеет официальный драйвер для семейства принтеров Z600 для пользователей Red Hat 9.0, но мы обнаружили полезное онлайн-руководство по переводу этого драйвера в форму, дружественную для Ubuntu (попробуйте сами, www.trodrigues.net/wiki/linux:ubuntu:lexmarkJx1290). Через десять минут использования командной строки мы уже печатали, а вот сканирование так и не заработало. Но и одна лишь возможность печатать на принтере Lexmark уже замечательна, хотя и требует некоторых усилий.

Цена: нет ($50)

Рейтинг: ★★

Модель на фото не является именно той, которая тестировалась



MP3 и доступ к сети

Жаждете ли вы получить онлайн-доступ, или классический Пинк Флойд, устройство должно «просто работать».

MP3-плейеры есть везде, и судя по числу затычек, торчащих в людских ушах, iPod'ов тоже хватает. Но позицию Apple по отношению к открытому ПО дружеской не назовешь – не считая некоторого вклада в HTML-движок для *Konqueror*, нам доступна ничтожная часть обширного ПО Apple. Думаете, поддержка iPod Linux'ом из-за этого имеет бледный вид? Отнюдь: iPod – это фиксированная платформа с большой пользовательской базой, и ручонки хакеров, конечно, до нее дотянулись, так что теперь у нас есть впечатляющая поддержка звука iPod под Linux. Но что она умеет делать?

iPod Video 80GB



Подключите ваш iPod, и почти сразу вылезет экран, предлагающий запустить *Rhythmbox* для управления библиотекой iPod'a. Да, наряду с подключением iPod'a к компьютеру в качестве просто устройства хранения данных, Linux также позволяет вам читать и писать музыкальную базу данных iPod'a посредством *Rhythmbox* – вы можете перетащить ваши файлы прямо на iPod и, если пожелаете, скопировать их оттуда. В силу того, что в Ubuntu есть встроенная поддержка MP3/AAC (и, конечно, всплывающий менеджер кодеков здесь вам поможет), вы сможете проигрывать музыку непосредственно с iPod'a. Недостает разве что поддержки видео, а также знания того, что делать с защищенной DRM-музыкой, заказанной через магазин Apple – можно только попытать удачи с *iTunes* и *CrossOver*.

Есть еще и другая возможность: установить Linux на ваш iPod, для большей гармонии с другими элементами вашей сети FOSS. Гениальные хакеры с www.ipodlinux.org/forums направят вас на нужные утилиты, поскольку сеть кишит проектами, касающимися запуска Linux на маленькой коробочке Apple, которые находятся на разной стадии завершения и успеха.

Цена: 12 500 руб.

Рейтинг: ★★★★

Беспроводная карта Belkin Wireless (PC Card)



В наши дни большинство ноутбуков – даже дешевых моделей – включают встроенную беспроводную сеть, но устройства PC Card все еще довольно полезны, если вы пытаетесь связаться сетью со старой машиной. Belkin – одно из наиболее известных имен в периферийном оборудовании, так что можно ожидать приличную поддержку Linux без больших трудностей с драйверами. Мы, однако, убедились, что ожидания и реальность часто не совпадают, когда дело касается оборудования и Linux...

Печально, но при вставке карточки происходит его величество Ничего. Ни звука, ни света – просто тишина и ни малейшего отклика из журналов ядра. Еще более печалит, что различные команды *pccardctl* тоже не дают никакой информации. Мы ожидали хотя бы чего-то вроде 'Card not recognised' (Карта не распознана) или некоторую информацию об устройстве, но утилита просто завершается, не делая ничего. Очень обидно. Единственной нашей возможностью оставалась перезагрузка со вставленным устройством, на случай, если у ядра есть проблемы с горячим подключением PC Card. При том, что тестовая машина была вполне стандартным ноутбуком Dell Inspiron, мы надеялись на большее!

Но не сработала даже перезагрузка: все еще не было никаких признаков присутствия устройства. В итоге, эта скучная сага открыла нам скверную сторону совместимости Linux с оборудованием: всем известно, что некоторые, в особенности экзотические, устройства не работают, но то, что с широко известной карточкой на широко распространенном ноутбуке ничего не происходит – это печальное состояние дел. Даже сообщение «Я прямо и не знаю, что это» было бы куда желаннее.

Цена: нет ($50–100, в зависимости от спецификации)

Рейтинг: ★

Samsung YP-Z5 4GB



Не все отовариваются в торговых точках Apple, но это не значит, что их следует оставлять на морозе. Мы добрались до первого неэппловского MP3-плейера, которого смогли найти, и вставили его в Ubuntu. Результат? Плейер был подмонтирован на компьютере для чтения и записи как обычное устройство хранения данных. Оно не появилось в *Rhythmbox*, что означает, что вам придется управлять вашей музыкальной коллекцией с помощью *Nautilus'a*. Однако вы можете проиграть вашу музыку непосредственно с устройства, как с iPod – даже суперпроприетарная WMA от Microsoft работает, если у вас установлены правильные кодеки.

Цена: 5500 руб.

Рейтинг: ★★★

Bluetooth-адаптер MSI Star Key

Начало было хорошим – ядро подцепило 'HCI USB' Bluetooth-драйвер, хотя никакой обратной связи от Gnome не было. Это было типичной проблемой во время тестирования: с помощью HAL и усиливающегося желания заставить устройства работать с Linux, мы надеялись на большую интеграцию с рабочим столом. То есть, после вставки устройства, вам нужно определить, что нужно сделать. Мы пытались установить сеть Bluetooth посредством Network Manager – но он не отобразил адаптер Bluetooth в списке используемых устройств [странная идея. Известно, что NetworkManager не поддерживает PAN-сети, по крайней мере, в настоящее время, – прим. ред.]. Как ни странно, в Ubuntu не установлен по умолчанию пакет *bluez-gnome*, и мы не обеспечены никаким графическим инструментом настройки. Тем самым устройство потеряло одно очко, ибо потребовалась дополнительная работа.

Однако после вторичной вставки адаптера в системном лотке Gnome появился значок Bluetooth. Правый щелчок мыши выдал нам меню Preferences, которое позволило задать тип устройства (настольный ПК или ноутбук), а также время невидимости. А больше ничего – только продолжительный поиск показал, что для принятия и отправления файлов требуется установить *gnome-bluetooth*. Ну почему его не включили в дистрибутив?

Тем не менее, заработав, устройство показало себя безукоризненным, посылая и принимая файлы без проблем на дистанции 20 м. Позор, что Ubuntu не полностью поддерживает Bluetooth прямо из коробки, ведь это надежное и компактное устройство – да и ценности у дистрибутива бы прибавилось.

Цена: 500 руб.

Рейтинг: ★★★



Устройства USB

USB расхваливали как радикальное решение всех проблем аппаратного обеспечения. Что даст наше испытание?

Здесь мы рассмотрим набор обычных USB-устройств тех типов, которые встречаются и используются вами каждый день. Изначально мы надеялись, что с помощью USB все будет работать при минимуме усилий, и все устройства определенного типа будут обслуживаться родовым драйвером. Таким путем дело не пошло, и хотя USB гораздо элегантнее, чем борьба между параллельным и последовательным портами, это не есть панацея. На сегодня лишь малый процент устройств упоминает Linux на коробке, причем обычно без всякой спецификации. Иногда можно догадаться о поддержке устройства Linux'ом, глядя на другие ОС: если на коробке написано, что оно работает с Windows и Mac OS X, а CD с драйвером не приложен – вероятно, это устройство общего типа, и под Linux тоже заработает.

Наушники Logitech 250 USB



В магазинах нечасто можно увидеть USB-наушники или головные телефоны – ведь почти каждый ПК имеет стандартные 3,5-мм аудио гнезда. Однако они могут пригодиться, если аудио-выходы заняты, скажем, записью проекта. Мы попробовали пару удобных мягких наушников от известного изготовителя Logitech.

Когда их вставляют, ядро дает очень мало информации, не считая обрывка фразы, что зарегистрирован новый драйвер интерфейса snd-usb-audio. Мы запустили *Beep Media Player* и... музыка заиграла через обычные колонки. Linux даже не попытался установить наушники в качестве устройства выхода, когда их подключили. Покопавшись в окне Preferences *Beep'a*, мы настроили модуль звукового выхода и смогли указать ему на USB-аудио устройство. Это хорошо для прослушивания музыки, но как насчет других программ? Утилита Sound Preferences в Ubuntu позволяет установить устройство выхода по умолчанию для некоторых программ, но не для всех, так что вам придется возиться с этим каждый раз, когда вы захотите переключаться с колонок на наушники. В принципе, наушники хорошие, но из-за этой мороки потеряли очко в рейтинге.

Цена: 1000 руб.

Рейтинг: ★★★

Web-камера Logitech



При подключении этого забавно выглядящего устройства Logitech, журнал ядра отметил новое 'video capture interface', но Gnome не предложил никакого содействия настройке. *Ekiga* запросила у нас 'video manager'. Выбор был между V4L, V4L2 (версии Video4Linux) или Picture. Информативно! Но когда мы выбрали V4L (стоит по умолчанию), *Ekiga* указала, что имеется включенный 'USB Video Class Device', а затем провалила тесты устройства.

Выбор V4L2 разрешил эту проблему. Тут главный экран *Ekiga* выдал раздражающе темное сообщение об ошибке: 'Your driver doesn't seem to support any of the colour formats supported by *Ekiga*. Please check your kernel driver documentation.' То есть, нет показа для этой web-камеры, и без малейшего намека на то, что есть какая-то возможность заставить ее работать.

Цена: от 300 руб., в зависимости от модели

Рейтинг: ★

Модель на фото не является именно той, которая тестировалась

Планшет Wacom ET-0405



Ни один рабочий стол художника, использующего компьютер, не обходится без графического планшета, и исторически Linux показал себя с хорошей стороны на этом направлении, поддерживая много простых устройств (хотя и не без трюков). Так что же случится, если подключить обычный Wacom к свежей установке Ubuntu? Журнал ядра говорит, что найдено 'Wacom Graphire and Wacom Intuos tablet', и перо тут же может использоваться в качестве мыши.

Устройство чрезвычайно чувствительно – малейшее надавливание производит эффект нажатия кнопки мыши: при использовании иногда обнаруживается, что вы «кликаете» и перетаскиваете за пределами графического редактора! Следовательно, перед тем, как рисовать, нужно аккуратно задать положение окна-холста, иначе вы рискуете закрыть другие окна или перетащить файлы в случайные каталоги.

Планшет без проблем сработался с *GIMP*; он был чутким и отзывчивым, и кнопки пера давали доступ к контекстному меню. Как и со многими другими устройствами, тестировавшимися в этой статье, обычно стоит держаться широко известных марок, даже если придется малость переплатить. Многие драйверы для оборудования для Linux пишутся разработчиками в свободное время, чтобы иметь поддержку своих устройств, и чем более популярно и распространено устройство, тем вероятнее, что оно заработает под вашим Linux.

Цена: нет ($50 б/у)

Рейтинг: ★★★★★

Web-камера Hercules Blog



Как и следовало ожидать, в тощем руководстве пользователя и на CD с драйверами (где имеется Windows Live Messenger) нет никакого упоминания о Linux. Эта 1,3-мегапиксельная камера рекламируется как «способная делать снимки для вашего блога» – странный ход, если учесть, что такое может любая цифровая камера! Она также поддерживает 'Vlogs' (видеоблоги) и возможность преобразовывать фильмы в форматы PSP и iPod.

При подключении ошарашенное ядро 2.6.20 немедленно выдало, что 'cannot get freq at ep 0x84'. Оно даже не сообразило, что это некое видеоустройство, не говоря уж о web-камере. Но с одними сообщениями ядра далеко не уедешь, и мы сделали попытку с *Ekiga*.

На этапе конфигурации устройства даже не было в списке. Ситуация с web-камерами под Linux, к сожалению, ужасна – даже несмотря на то, что многие устройства базируются на одинаковых чипах. Тот факт, что пользователи вынуждены иметь дело с V4L и разбирать непонятные сообщения об ошибках, иллюстрирует острую нужду в улучшениях. Если вы идете покупать web-камеру, введите ее имя или номер модели вместе со словом Linux в Google и посмотрите, заработала ли она хоть у кого-нибудь!

Цена: нет ($40)

Рейтинг: ★

Внешний DVD-привод HP DVD940e Lightscribe



Большинство пишущих DVD-приводов успешно работают под Linux – по крайней мере, по части прожига. Устройство Lightscribe от HP, рассмотренное в LXF91, не просто штампует DVD: оно позволяет создавать этикетки и рисунки на поверхности специальных DVD-дисков. Незачем терять время и возиться с наклейками, неуклюже располагая их на поверхности диска – в процессе записи создается рисунок на нерабочей стороне DVD.

После подключения устройства ядро опознает его как устройство 'scsi generic', хотя Ubuntu отмалчивается, и никакое пишущее приложение не выскакивает. Тем не менее, функции записи дисков *Nautilus*'а работают хорошо – просто перетащите файлы в папку **CD/DVD Creator**, нажмите Write To Disc. *Nautilus* выкатит список задания скорости записи, позволяющий добиться наибольшей скорости нашего записывающего устройства: 6.2x для записи DVD и 56x для CD.

Быстрый и бесшумный, DVD940e – это убедительный выбор, если вам нужен внешний DVD-«резак»; если вы предназначаете его лишь для резервных копий, возможность делать этикетки может показаться неуместной роскошью. HP поставляет две программы для создания этикеток (от Lite-On и LaCie), плюс «системный» пакет, обеспечивающий библиотеки для обеих. Досадно, но эти пакеты доступны только в формате RPM – перевести их в Deb-формат с помощью Alien не велика трудность, но зачем нам лишний шаг? Потребовалось также установить *libpng3*.

При пользовании программами этикеток они ведут себя приемлемо; они довольно элементарны с угловатым Motif-подобным GUI [что, вообще говоря, странно – *LaCie LightScribe Labeler* базируется на *Qt*, – прим.ред.], но легко осваиваются. Будь сопутствующие программы доступными в форматах, отличных от RPM, мы несомненно поставили бы ему 5, но дополнительные затраты снижают оценку на одно очко. Как бы то ни было, это превосходный DVD-резак.

Цена: 2500 руб.
Рейтинг: ★★★★

КПК Tapwave Zodiac 2



Dekaritae

Устройство от Tapwave больше похоже на портативную игровую консоль, чем на традиционный КПК: оно щеголяет аналоговым джойстиком слева и набором кнопок справа. Если вы не хотите походить на человека, играющего в N-Gage во время совещаний, поверните дисплей на 90 градусов и держите агрегат вертикально.

На Zodiac'е запущена PalmOS, хотя и с собственным GUI, но учитывая, что PalmOS на сцене уже несколько лет, пора ожидать некоторого успеха с Linux.

Мы опустили устройство в его «кроватку» и нажали кнопку HotSync, в надежде, что какая-нибудь из утилит Gnome вступится и спросит нас, что мы желаем перенести. Облом: ядро сообщило, что присоединено USB-устройство, но не вызвало ничего другого. Эх, похоже, что настраивать придется самостоятельно.

Системное меню Ubuntu содержит *gnome-pilot*, который сперва предлагает диалог-мастер выбора устройства. Альтернативы – последовательный порт либо USB; мы выбрали последнее и получили весьма детальные подсказки, например, про узел и скорость устройства. Мы кликали, принимая стандартные установки, пока не дошли до HotSync. Угадайте, что было дальше? Правильно, сообщение об ошибке: 'Failed to connect... Check your configuration, as you have requested old-style usbserial `ttyUSB` syncing...' (Не могу соединиться... Проверьте свою конфигурацию, так как вы запросили старый usbserial `ttyUSB`), и дальше про модули ядра.

Ну не глупо? С таким сообщением об ошибке не вдруг разберется и бывалый линуксоид – что уж говорить о новичках. Учитывая, что этому устройству почти два года, и оно основано на широко известной PalmOS, можно было надеяться на лучшую поддержку, чем предлагаемую в Linux на настоящий момент. И ожидать большего от такого (предположительно) развитого приложения, как *gnome-pilot*. Жаль, что Zodiac 2 – вполне приличное устройство – больше не производится.

Цена: нет ($140 б/у)
Рейтинг: ★

Web-камера Genius Look 316



Маленький логотип на коробке заявляет о поддержке Linux. Однако просмотр установочного диска выявляет скрипт и один скомпилированный бинарный драйвер… для ядра 2.6.5. Благодарим покорно: теперь что, нужно вернуться назад в 2004 и пробовать его там? Вот так: ни исходного кода, ни подсказки, что делать, если сдуру обновил ядро в последние три года. Сайт www.geniusnet.com не дал ничего полезного. Техническая спецификация оговаривает именно ядро 2.6.5, и по наивности можно решить, что это значит 2.6.5 и выше.

Даже открытые драйверы не смогли заставить устройство работать, фактически подвесив несколько ПК с Ubuntu! Печальная история повторялась, пока мы не попробовали Fedora Core 6, используя RPM'ы с http://mxhaard.free.fr для добавления драйвера *spca5x* к современному ядру. Хотя ничего больше не слетало, обычный круг подозреваемых (*Xawtv*, *MPlayer* и т.д.) не смог работать с устройством напрямую, и только некоторые утилиты (например, *Ekiga* и *CamE*) вели себя хорошо. Для использования Look 316 в качестве web-камеры вам следует взять *Ekiga* или ПО *SPCA* с того же сайта, где вы достали работающие драйверы. *SPCA* умеет делать экранные снимки, работая как сервер изображений (*CamE* тоже годится), и дает вам (причудливый) GUI-интерфейс ко всем функциям камеры.

Что вы действительно получаете, так это возможность пропихивать, на 15 кадрах в секунду, VGA-изображения через стандартный интерфейс USB 1.1. Устройство заявлено на скорость 30 кадров в секунду на QVGA, но то ли из-за используемых утилит, то ли из-за драйвера мы её не достигли. Упаковка обещала захват статичного изображения на 1,3 мегапикселях, но само устройство ограничено разрешением VGA – изображение «высокого разрешения» получается дымом и зеркалами DirectX под Windows (а можно просто повысить качество рисунка в *GIMP*). Ах да, мы забыли сказать про звук: его нет.

Цена: 650 руб.
Рейтинг: ★★





Игровые контроллеры

Как справится Linux с простейшей типовой игровой панелью и с более экзотическими устройствами?

Игры под Linux – это не частое явление, но благодаря *Wine* и *Cedega* у нас есть десяток имен, известных под Windows. В сети также существует процветающее сообщество эмуляторов, охватывающих классические консоли и аркадные машины. Если у вас слабость к лихим гонкам, имитаторам полета или новым консолям, вам действительно нужен внешний контроллер. К счастью, прошли те дни, когда установка джойстика означала долгие пляски вокруг ужасных портов звуковой карты. Сейчас почти каждый продаваемый игровой контроллер присоединяется через USB, и существует такая вещь, как «типовой» игровой пульт USB (дешевая штука, предлагаемая игровыми магазинами).

Типовой игровой пульт USB

В большинстве игровых магазинов вы можете выбрать базовый игровой пульт USB за 10 с небольшим долларов – обычно за бренд сходит название магазина, а иногда пульт настолько типовой, что вообще не имеет бренда. Им обычно недостает возможностей аналогового контроля, но они достаточно хороши для игры на консольных эмуляторах. Мы тестировали пульт PC Line USB (от GAME), имеющий D-пульт и восемь кнопок в формате SNES/PlayStation. Ubuntu не выказал никакой реакции при подсоединении устройства, но быстрое сканирование с помощью *dmesg* подтвердило, что ядро его опознало.

Игры также не имели проблем с определением игрового пульта. В некоторых играх или эмуляторах нужно переключить ввод с клавиатуры на игровой пульт и, возможно, настроить значения кнопок, но работы здесь не больше, чем было бы под Windows. Дешевые бросовые панели выглядят убогими рядом с аналоговыми джойстиками и рулевым управлением с обратной связью, но они идеально подходят, если вам захотелось снова поиграть в *Sonic the Hedgehog*, и что самое приятное, они работают под Linux!

Цена: нет ($10–20)

Рейтинг: ★★★★★

Джойстик Logitech WingMan Extreme



Этот аналоговый монстр щеголяет 12 кнопками на всем устройстве, плюс D-пульт сверху и регулятор сбоку. Очевидно, что он разрабатывался для симуляторов полета и авиационных игр, но как он уживается с классической Linux'овской авиапотехой, *FlightGear*? Как и в случае с типовым пультом, видимых фанфар подключение джойстика не вызвало, однако журнал ядра показал, что обнаружено устройство ввода 'Logitech Inc. WingMan Extreme Digital 3D'. Игры, основанные на SDL, не имели проблем при работе с этим устройством.

Джойстик превосходно распознался в *FlightGear* – никакой настройки не потребовалось. Все кнопки работали, при D-пульте, контролирующем движение глаз пилота в кабине, и аналоговый рычаг был удивительно точен. Идеально работал даже скользящий регулятор. Это впечатляющее устройство – прочно разработанное и отзывчивое.

Бесспорно, успехом в Linux этот джойстик обязан бре нду: Logitech широко известен, и, следовательно, его продукты наиболее вероятно будут поддерживаться Linux. Если вы собрались купить контроллер имитатора полета, но не можете найти именно эту модель, мы настоятельно рекомендуем держаться известных марок – использование устройств от мелких изготовителей или совсем без бренда чревато неприятностями.

Цена: 800 руб.

Рейтинг: ★★★★★

Игровой пульт PS2 с USB



Не можете оторвать рук от привычного консольного игрового пульта? Можете использовать его под Linux, при условии, что найдете правильный USB-адаптер. Попробуйте поискать в Google и eBay – вам выдадут USB-адаптеры для всех видов контроллеров, включая SNES, N64 и игровые пульты от PlayStation. Прискорбно, однако, что здесь Linux имеет несколько камней преткновения. Мы попробовали преобразователь PlayStation 2 на USB от Play.com, подключив главный игровой пульт PS2. Ядро опознало его, программы – тоже. Но что-то было глобально неправильно с поддержкой аналогового рычага в этом релизе Ubuntu: преобразователь постоянно посылал назад сообщения о движении рычага даже когда они были центрированы, таким образом лишая возможности переобозначить кнопки в ZSNES. В *Planet Penguin Racer* рычаги передавали кое-какие движения Туксу, хотя и в чрезвычайно неприятной манере, какими-то толчками. В итоге мы вынуждены поставить данному конкретному преобразователю отрицательный рейтинг, но если вам нравится идея подсоединения консольного пульта, не отчаивайтесь. Многие пользователи Linux сообщали об успехах с устройствами SmartJoy и Boom (хотя функция rumble пока не работает), так что поглядывайте на eBay.

Цена: недоступно

Рейтинг: ★

Рулевое колесо Thrustmaster RGT



Более экипированных контроллеров не существует: мощное рулевое колесо с обратной связью, содержащее D-пульт, 7 кнопок, постепенные и последовательные рычаги за колесом, и вдобавок рукоятка передач. Для ног имеется большое трехпедальное приспособление, присоединенное к основному устройству. Эта штука имеет даже собственное питание!

При USB-соединении все было нормально: подключите его и, как для всех тестировавшихся здесь пультов, ядро выдает короткое указание о том, что устройство обнаружено. Нашим первым тестом был *Torcs*, один из самых известных имитаторов вождения под Linux, который определил руль, но имел большие проблемы при взаимодействии с ним. Фактически, работала только педаль тормоза – рулевое колесо центрировалось при движении, а акселератор был абсолютно неисправным.

Как ни грустно, *Torcs* не дал шансов исправить это во время игры. Мы достигли большего успеха с *Planet Penguin Racer* (бывший *TuxRacer*), при этом средняя педаль функционировала как акселератор, а руль имел отдачу на крутых поворотах. Рулить пингвином кажется немного странноватым; но, по крайней мере, видно, что рулевое колесо в порядке. Thrustmaster – это, пожалуй, перебор, учитывая малочисленность высококачественных игр-гонок под Linux, однако если вы ищете полноценное рулевое колесо, мы рекомендуем потратиться на него.

Цена: 9000 руб.

Рейтинг: ★★★★

Вердикт

Список пожеланий составлен? Надеемся, результаты наших тестов хоть немного да воодушевили вас…

Если усреднить счета всех устройств, получится, что средний рейтинг примерно равен 3 – среднестатистическое устройство заработает под Linux, если вы не побоитесь некоторых упражнений с командной строкой. Мы уверены, что смогли бы добиться лучшего показателя, будь мы разборчивее с тестируемым оборудованием, но это не было целью нашего теста. Обстрел картечью, использованный нами, означает, что мы выбирали не то, что работает хорошо, и даже не то, что работает хорошо с некоторыми вывертами. У вас есть необработанный, бесцензурный список того, что работает, а что нет.

Было бы очень просто закончить статью, бодро возгласив, что поддержка драйверов Linux становится все лучше и лучше, но так ли это? Мы существуем уже почти 16 лет, начиная с тех времен, когда еще не была выпущена Windows 3.1. Если число пользователей Linux не превысит некоторую критическую массу за следующие пять лет, реальное изменение ситуации с драйверами маловероятно.

Время менять мир

Это не значит, что все потеряно. Надеемся, вы усмотрели из представленных тестов несколько тенденций:

» Более распространенные торговые марки поддерживаются лучше. В высшей степени рекомендуются, в частности, Hewlett-Packard и Nvidia.

» Покупка новехонького оборудования грозит неприятностями. Аналогично, покупка очень старого оборудования вряд ли обойдется безболезненно. Лучшим выбором будет то, чей возраст по крайней мере шесть месяцев (то есть время обновления дистрибутива), но меньше лет пяти.

» Если вы не боитесь командной строки, ваши шансы на успех удваиваются.

Мы все еще далеки от подхода в стиле OS X «подключите и работайте», но мы чуточку ближе к стилю Windows «plug and pray [подключите и молитесь]». Если вы хотите улучшить поддержку оборудования под Linux, есть три способа сделать это:

1. Написать драйвер ядра
2. Написать инструкцию по приведению устройства в рабочее состояние
3. Проголосуйте кошельком

Если вы пока что не хакер ядра, пункт 1 не привлекает. Если вас страшит мысль провести часы в мучительных настройках, пока устройство каким-то чудом не заработает, пункт 2 не для вас. А вот пункт 3 может выполнить каждый. Если есть устройство, которое вы хотели бы приобрести, но оно не поддерживает Linux, пошлите электронное письмо изготовителю и объясните, почему вы выбираете продукцию их конкурентов. HP и Nvidia такие белые и пушистые неспроста – они имеют команды разработчиков, занимающиеся написанием драйверов для Linux, и обе присылают множество исправлений в ядро. Нам нужно поддержать их, чтобы показать, что совместимость Linux с оборудованием значит для нас многое. LXF

Обратная связь

У вас был какой-нибудь плохой или памятно хороший опыт обращения с оборудованием под Linux? Какие изготовители особенно дружелюбны, или вы считаете, что заслуживает похвалы отдельное web-сообщество, которое вам помогло? Пошлите нам письмо на letters@linuxformat.ru и выразите ваши чувства, пошлите письмо на answers@linuxformat.ru, если у вас есть проблема с оборудованием, и вы не можете разрешить ее, даже после поиска по соответствующим форумам, и, конечно же, вам всегда рады на живом обсуждении в форумах на www.linuxforum.ru.





В защиту GPL 3

Illustration image: istockphoto.com

Предложения по GPL 3 не раз попадали под огонь критики. Но **Пол Хадсон**, побеседовавший с **Ричардом Столлмэном** и **Брюсом Пиренсом**, считает, что ее нужно поддержать.

***«GPL нарушает принцип Наполеона: никогда не мешай своему врагу совершать ошибку»* Эрик Рэймонд**

Критиков GNU GPL хватает как внутри сообщества, так и вне его. Но благодаря GPL у нас есть ядро Linux и инструменты GNU. Восемь из десяти самых загружаемых приложений на SourceForge лицензированы GPL. Ее же применяют около 66% проектов на Freshmeat, 68% проектов на SourceForge, 40% приложений из стандартной установки Fedora и десятки других известных проектов, среди которых *MySQL*, *Qt*, *Gimp*, *Inkscape*, *Scribus*, KDE и *Evolution*.

Так зачем же нам новая лицензия?

Когда Free Software Foundation (FSF) готовила масштабные изменения к GPL 2 в виде GPL 3, многие наблюдатели решили, что организация «сеет смуту», в лучшем случае – добавляющую забот небольшим проектам, а в худшем – привносящую политику и устаревающие со временем технологические аспекты в простую лицензию, которая служит сообществу почти 20 лет. «Когда была написана лицензия GPL», отмечает легенда Debian и обозреватель открытого ПО Брюс Пиренс, «не было Интернета, музыку записывали на пластинки, видео – на пленку, а вместо DRM были примитивные программные средства защиты от копирования». У критиков найдется что на это возразить? Разве не время спросить, насколько новая лицензия будет полезной сообществу?

> «С помощью GPL 3 мы пытаемся спасти свободное ПО от судьбы, которая хуже смерти.» Ричард Столлмэн

На самом деле, Пиренс считает почтенный возраст GPL недостатком, а не достоинством. Как и я, он является сторонником GPL 3. Я считаю, что важные поправки в GPL 3 дают ей большой потенциал, и она может поддержать сообщество и гарантировать его будущее так, как никогда не смогла бы ее предшественница.

Драматические изменения

***«Самый частый протест против GPL 3, который я слышу: «GPL 2 и так неплоха!» Но GPL никогда не была автономной лицензией. Она всегда зависела от интерпретации авторского права на местах и других законов, которые придавали ей силу. Все это со временем изменилось»* Брюс Пиренс**

Прошло полтора года с того дня, когда я встречался с Эбеном Могленом [Eben Moglen], чтобы обсудить планы по созданию и утверждению GPL 3 (тогда он был генеральным юрисконсультом FSF). Он думал, что FSF просто введет в действие новую лицензию в начале 2007 года после двенадцати месяцев публичного обсуждения и критики. Сейчас начало июня, а окончательной редакции GPL 3 все еще нет [GPL 3 увидела свет 29 июня 2007 года, – прим. ред.]. Но, вообще говоря, запаздывание составляет несколько месяцев, и с учетом объемности обсуждений GPL 3 это не так уж плохо. Фундаментальная цель Столлмэна «защитить свободу каждого пользователя» не изменилась – изменились лишь методы обеспечения этой защиты. Вот основные положения:

» **Ни одно из лицензированных по GPL 3 произведений** не может быть «эффективной технологической мерой» для запрета доступа к данным

в соответствии с определением DMCA [Digital Millennium Copyright Act, Закон о защите авторских прав в цифровую эпоху, 1998, – прим. перев.], EUCD [European Union Copyright Directive, Европейская директива о защите авторских прав, 2001, – прим. перев.] и другими международными соглашениями. Дистрибутивы GNU/Linux имеют давнюю проблему с чтением DVD-дисков, потому что последние защищаются системой CSS (Content Scramble System). И хотя взломать CSS было очень просто, она классифицируется как эффективная технологическая мера для защиты содержимого диска от копирования, и действия по ее взлому незаконны во многих странах. GPL 3 решает эту проблему так: «передавая охраняемое произведение, вы отказываетесь от всех юридических прав запретить обход технологических мер» – например, вы не можете использовать программу, лицензированную по GPL 3, чтобы запретить пользователям доступ к содержимому DVD-дисков и других носителей.

» **Не должно накладываться никаких ограничений на изменение продуктов GPL 3**. Телевизионная приставка TiVo, управляемая GNU/Linux, содержит аппаратный механизм, проверяющий цифровую подпись TiVo в ядре Linux перед запуском устройства. Таким образом, мы можем изменить исходный код ПО приставки (ядра Linux), но не сможем использовать модифицированное ПО, так как оно не содержит цифровой подписи TiVo. Одной из целей GPL 3 является запрет этой так называемой «тивоизации» с помощью требования, согласно которому при изменении любого лицензированного по GPL 3 компонента системы остальные DRM-механизмы не должны препятствовать ее работе.

» **Наконец-то включены патенты на ПО**. Каждый, кто вносит свой вклад в программу, должен предоставить «неэксклюзивную всемирную и безвозмездную патентную лицензию на существенные патентные притязания в сделанном вкладе, регулирующую производство, использование, продажу, предложение о продаже, импортирование и другое применение, изменение и распространение этого вклада». На первый взгляд каждый, кто изменит одну строку кода, должен предоставить лицензию лишь на те патенты, которые затрагивает эта строка, но в другом разделе GPL 3 говорится: «Вы должны лицензировать произведение полностью, как единое целое». Это означает, что при изменении исходного кода вы фактически создаете собственную версию, копируете 99% продукта у исходных разработчиков, а затем добавляете свой код. Все это становится вашим собственным произведением и подпадает под действие положения о патентах.

Никаких соглашений о патентной защите по типу Microsoft и Novell в будущем. Этот раздел новой лицензии вызывает самые ожесточенные споры. Отчасти потому, что соглашение Microvell полностью соответствует условиям GPL 2 (хотя и противоречит ее духу), отчасти из-за того, что пока – по крайней мере, на момент написания статьи – не решено, датировать ли лицензию так, чтобы все сделки по типу



› **Эбен Моглен считает, что GPL 3 подхватят многие компании, которые до этого совсем не использовали GPL, и это может принести Microsoft серьезные убытки.**

James Duncan Davidson

Microvell были недействительными или же установить некий стартовый момент и, таким образом, «выгородить» Novell. Эта часть лицензии является наиболее важной. Как сказал мне Ричард Столлмэн: «С помощью GPL 3 мы пытаемся спасти свободное ПО от судьбы, которая хуже смерти: стать фактически несвободным за счет владельца патента, который заставляет пользователей платить за использование прав, которые им дает GPL».

Включение этих важных изменений значительно увеличило размер лицензии: если в GPL 2 менее 3000 слов, то в GPL 3 – 5324 слова. И далеко не все согласны с данными изменениями.

Противники

***«Я не думаю, что мы должны препятствовать попыткам GPL сделать открытое ПО проприетарным, это окажется вредным нам самим и глупым»* Эрик Рэймонд**

Заметная часть критики в адрес GPL 3 исходит от сообщества разработчиков ядра Linux. Особенно усердствовал Линус Торвальдс [Linux Torvalds]. Во всем этом есть доля иронии, поскольку сложно говорить, что Linux лицензирован только по GPL, хотя в его коде и утверждается обратное. Моглен сказал мне, что у него есть «ясное ощущение, что лицензия ядра Linux – это не GPL и только GPL. Тому есть две причины. Во-первых, за время жизни ядра предлагались различные методы включения исправлений и взаимодействия с двоичными модулями – но разработчики не пытались прибегнуть к GPL. Я рассматриваю это как попустительство к действиям, которые можно было запретить.

Кроме того, в ядре есть GPL-символы и механизм связывания (taint), так что технически API подсказывает существование двух типов несвободных модулей: допустимых и недопустимых.»

Частично, данная проблема возникает потому, что немногие знают, чего нельзя, и что можно. В результате бинарные модули ядра отделяются от дистрибутивов, и пользователи должны устанавливать их сами. Например, код драйвера Nvidia компонуется с GPL-ядром, что идет вразрез с условиями GPL 2. Как заметил Брюс Пиренс, «Линус сказал много непонятных слов об этой проблеме, но ни одно из них не может считаться юридическим оправданием подобной практики. Каждый раз, когда он говорит что-то новое по данной теме, он еще больше запутывает дело. Я думаю, некоторое время назад он это понял и остановился».

Если ядро могло стать настолько успешным без буквального следования стандартной интерпретации GPL 2, то зачем нужна GPL 3? Вы »



Licensed by John Anderson under GPL

› **Брюс Пиренс напомнил мне, что GPL-лицензированное ПО заменило Taligent, Project Monterey и другие недешевые «недострои».**



National Retail Federation

› Исполнительный директор Microsoft Стив Баллмер на ежегодной конференции Национальной федерации розничной торговли в 2001 году назвал Linux «раком интеллектуальной собственности, который прилипает ко всему, чего касается». С того времени Microsoft выпустила набор сервисов *Microsoft Windows Services for Unix*, содержащий GPL-лицензированный код.

» можете даже спросить, какой вообще смысл в лицензии, если разработчики ядра не придерживаются единого мнения относительно компоновки свободного кода с несвободными модулями?

Опрос, проведенный в сентябре 2006 года среди 29 лиц, внесших наибольший вклад в ядро Linux, показал, что 28 из них против перехода на GPL 3. Двадцать девятый воздержался. По мнению двадцати трех из этих двадцати девяти, GPL 3 гораздо хуже GPL 2, потому что уничтожает всякую возможность движения. Грег Кроа-Хартман [Greg Kroah-Hartman], также принимавший участие в опросе, дал лицензии оценку «-3». Вот как он прокомментировал эту оценку: «Я бы не стал применять v3 (она в самом деле мне не нравится, и моя компания имела бы большие проблемы, разрешив мне попользоваться ее черновиком)». Когда я спросил его мнения о результатах голосования, он сказал: «Основной вывод, который я хотел бы подчеркнуть: многим нравится положение о DRM, а некоторые даже считают, что публиковать ключи необходимо и в рамках GPL 2, но они уверены, что технологические спецификации нельзя включать в официальный документ, который должен прослужить не один год».

Таким образом, для Кроа-Хартмана проблема в том, что DRM нет места в GPL, а не в том, что положение о DRM – это плохо. Похоже, Торвальдс придерживается того же мнения, хотя выражает его более резко: «В моем мировоззрении нормально, что другие люди делают глупости. Можно сожалеть об этом, но, в конце концов, это их выбор».

Microsoft и Novell

«Novell и Microsoft создали лазейку в законе. FSF может милостиво согласиться препятствовать применению подобных соглашений к ПО, лицензированному по GPL 3, лишь после мая 2007 года, что позволит Novell распространять программы, использующие GPL 3»
Брюс Пиренс

Ненадолго отложив DRM в сторону, мы находим в GPL 3 другой спорный вопрос, отстоять который будет еще сложнее: должны ли допускаться соглашения, подобные Microsoft и Novell? Просто чтобы освежить вашу память, напомню, что по этому договору Microsoft, среди прочего, обеспечивает защиту клиентов Novell от патентных тяжб со своей стороны в обмен на процент от продаж ПО Novell с открытым исходным кодом. И какие бы доводы не приводила Novell, очевидно, что Microsoft использует это соглашение, чтобы сказать потенциальным потребителям GNU/Linux: выбирайте Novell, если не хотите столкнуться с правовыми проблемами в будущем. По существу, это делает лицензированное по GPL ПО менее свободным, поскольку вы вынуждены использовать решения Novell, если хотите избежать беспокойства.

Дрожащие корпорации

Недавно TiVo и Novell составили свои годовые отчеты для USSEC (Комиссии США по ценным бумагам и биржам). Обе компании отметили, что принятие GPL 3 может вызвать проблемы в будущем. В части, TiVo утверждает: «Если предложенная на настоящий момент версия будет принята, мы можем потерять возможность включать операционную систему GNU/Linux в наше программное обеспечение, что неблагоприятно отразится на нашем бизнесе». Novell беспокоит то, что ее соглашение с Microsoft может оказаться под угрозой, потому что MS не станет гарантировать патентную неприкосновенность GPL-проектов. Нет, мне не жаль ни одну из этих компаний. Напротив, хорошо, что GPL 3 оказывает запланированное действие.

Здесь есть одно препятствие: соглашение между Novell и Microsoft полностью соответствует условиям GPL 2, а FSF не хочет, чтобы оно соответствовало GPL 3. В окончательной редакции лицензии будет пункт, запрещающий подобные соглашения, но он датирован и относится только к договорам, заключенным после появления GPL 3 [фактически, 28 марта 2007 года, – прим. ред.]. Это означает, что договор между Novell и Microsoft сохраняет свою силу.

Но даже если компании, использующие такие соглашения, явно исключаются из правового поля новой лицензии, это не означает ничего хорошего для остальных фирм. Поскольку FSF владеет авторскими правами на значительную часть любого дистрибутива GNU/Linux, она может изменить лицензию для этого ПО так, как посчитает нужным. GPL 3 же явным образом нацелена на две компании, хотя их действия полностью легальны. Я не говорю, что согласен с ними, но могут ли компании, которые планируют использовать GPL-лицензированное ПО, быть уверенными в том, что FSF в очередной раз не изменит лицензию, потому что ей не понравятся их действия? Хотя третья редакция GPL обошла стороной соглашение между MS и Novell, другим фирмам может так не повезти. Должна ли возможность изменения лицензии беспокоить компании? Столлмэн так не считает: «Зачем отказываться от использования GPL 3 только потому, что вам может не понравиться GPL 4?». «Это как сжечь свой дом из-за боязни, что пожар может случиться в следующем году».

Сражаясь за GPL

«Принимая во внимание неумение FSF изящно признать победу, я, честно говоря, думал, что разработка третьей версии превратится в мрачный оппозиционализм» **Эрик Рэймонд**

DRM, патентные соглашения и компоновка с бинарными модулями противостоят GPL, и у FSF могут возникнуть проблемы с ее продвижением в ряде областей. Однако, вопреки ожиданиям Рэймонда, FSF оказалась удивительно отзывчивой организацией. На ранних стадиях обсуждения были внимательно выслушаны мнения всех заинтересованных сторон, и в более поздних редакциях FSF старалась пойти на уступки.

На самом деле, у некоторых из названных проблем есть решения. Хотя DRM нельзя реализовать в GPL-программах, все еще есть четыре области, где его использование законно. И, по мнению Пиренса, именно они являются «лучшими, более безопасными и надежными, независимо от лицензии». Сюда относится аппаратное обеспечение, сопроцессор, микроядро, которое управляет настоящим ядром и программы, работающие в режиме пользователя. Если ядро Linux будет переведено на GPL 3, использование DRM в любом из этих мест будет полностью законным. Однако производители ПО приняли эти методы с большой неохотой даже для GPL 2. Например, многие антивирусы требуют установки двоичных модулей ядра, хотя это и не обязательно для решения их задач. Конечно, если люди сознательно нарушают GPL 2, то почему с выходом третьей версии что-то должно измениться?

Другая проблема, которую следует упомянуть, связана с переводом существующего кода на GPL 3. В большинстве проектов, лицен-

зированных по GPL 2, используется формулировка «вторая версия Лицензии или более поздняя версия по Вашему выбору». Но на практике разработчикам не так-то просто перейти со второй версии на третью, поскольку это требует согласия всех авторов исходного кода проекта. Один из самых крупных GPL-проектов – ядро Linux – может потребовать множества усилий. Доктор Джеймс Боттомли [Dr James Bottomley], один из основных участников проекта, говорит об этой задаче как о «практически невыполнимой». Ларс Линднер, [Lars Lindner] один из разработчиков Liferea, признает, что его проект не перейдет на GPL 3 именно из-за перспективы связи с каждым участником и получения его согласия на принятие новой лицензии. Итак, означает ли это, что ядро Linux и другие проекты навсегда связаны с GPL 2 просто потому, что очень трудно опросить всех разработчиков? Нет – мы можем пойти другим путем и переписать эти участки кода самостоятельно. «Изменить лицензию гораздо проще, чем думают многие, – утверждает Пиренс. – Разработчики ядра никогда не отказывались от замены кода там, где это необходимо, и никогда не медлили с работой. Неважно, каков был ее объем. Достаточно посмотреть на замену *BitKeeper*'а *Git*'ом: огромная работа, которая потребовала полного переписывания кода и, тем не менее, была завершена за 5 недель. Таким же образом можно поработать и над кодом, принадлежащим противникам GPL 3».

Переход

***«Добровольные разработчики не хотят быть бесплатными сотрудниками некой проприетарной компании»* Брюс Пиренс**

Никто до конца не уверен, какие программы будут переходить на GPL 3, а какие нет. Похоже, проекты FSF, такие как GCC и Bash, перейдут на GPL 3 вскоре после выхода ее окончательной версии, в то же время другие проекты еще нужно убеждать это сделать [во время подготовки русскоязычной версии этой статьи стало известно, что *Samba 3.2* и выше будет распространяться по GPL 3, – прим. ред.]. Если Пиренс и другие правы в том, что код можно переработать, то руководителей проектов можно было бы склонить к этому решению. Им действительно нравится GPL 3, и они хотят выпускать ПО под этой лицензией. Даже Эрик Рэймонд [Eric Raymond], решительный противник GPL вообще, считает, что лучше предотвратить раскол GPL 2/3. «Думаю, что у GPL 3 есть возможность быть более подготовленной, чем GPL 2, – говорит он. – Если мы собираемся использовать copyleft [по определению FSF, «основной метод, позволяющий сделать программы или другие работы свободными», см. **www.gnu.org/copyleft**, – прим. ред.], механизм этого процесса должен быть продуман как можно лучше, а сообщество – быть максимально сплоченным» Разработчик Liferea Ларс Линднер сказал мне: «Для совершенно новых проектов я бы использовал GPL 3». Похожий ответ я получил от Доминика Маззони [Dominic Mazzoni] из команды разработчиков *Audacity*: «Мы не собираемся изменять лицензию *Audacity*, потому что не запрашивали у разработчиков передачу авторских прав. Вряд ли мы сможем связаться со всеми, и даже если сможем, я уверен, что один-два будут против использования GPL 3... А вообще я думаю, что GPL 3 будет успешной, ее могут принять от половины до двух третей всех проектов, лицензированных сейчас по GPL 2.»

Итак, треть проектов продолжит использовать GPL 2 либо потому, что они не могут сменить лицензию на GPL 3, либо из-за несогласия с какой-то ее частью. Например, Джеймс Боттомли говорит: «Условия антитивоизации с философской точки зрения противоречат моему взгляду на открытое ПО». Другие же проекты хотят посмотреть, на что будет похожа финальная версия лицензии. Марк Хинкл [Mark Hinkle], вице-президент по развитию бизнеса и связям с сообществом в компании Zenoss, создателе очень популярного инструмента управления *Zenoss Core*, говорит: «Мы пристально следим за тем, когда выйдет финальная версия GPL 3... Однако, как участники сообщества, мы стремимся максимально дополнять другие проекты, которые мы поддерживаем и используем, а также принимать решения, оптимальные для наших клиентов».



› GPL – это лицензия-вирус? Ричард Столлмэн так не считает: «Программа не может подхватить GPL, просто находясь на одном компьютере с другой программой, которая использует GPL, – говорит он. – GPL действует лишь когда вы используете защищенный этой лицензией код как часть большой объединенной программы. Вместо того, чтобы называть GPL «вирусом», нужно называть ее «похожей на мухоловку» (spider-plant-like)». [мухоловка – хищное насекомоядное растение, – прим.ред.]

Это означает, что *Zenoss Core* может просто продолжить использовать GPL 2, если не будет убедительных причин менять версию. И это очень грустно, потому что показателем успеха GPL 3 будет число проектов, которые перейдут на нее – не из желания поддержать, а по практическим соображениям. Главное различие между GPL 3 и GPL 2 в том, что шансы первой на успех тем больше, чем больше людей ее выбирают.

Я считаю, это самый сильный довод в пользу широкого распространения GPL 3. Если никто за пределами FSF не будет использовать GPL 3, тогда ее пункты о патентах будут бесполезными, ее положения против DMCA будут проигнорированы и ничего не изменится. Чем больше людей внесут свой вклад в проекты, лицензированные по GPL 3, тем больше программ будут защищены новыми пунктами о

«Главное различие между GPL 3 и GPL 2 в том, что шансы первой на успех тем больше, чем больше людей ее выбирают».

патентах и тем безопаснее мы сможем себя чувствовать. Да, переход на эту лицензию сложно осуществить, но взамен этой работы мы получим гарантированную свободу ПО на годы вперед – GPL 2 могла только обещать эту свободу при некоторых условиях. За это несомненно стоит бороться! LXF

Ваши пять



А что думаете вы: надеетесь перевести свою программу по GPL 3 или считаете, что эту лицензию не стоит принимать? Напишите нам по электронной почте **letters@linuxformat.ru** или подключитесь к дискуссии на **www.linuxforum.ru**...



Тестируем службы поддержки

В самом деле, хорошо ли работают службы поддержки известных дистрибутивов Linux? Управляющий отделом IT **Дэвид Картрайт** решил притвориться новичком и проверить.

Об авторе

Дэвид Картрайт – консультант в сфере IT, он занимается проектированием, установкой и сопровождением корпоративной IT-инфраструктуры для средних компаний. Его последние должности – IT-директор британского офиса одного из международных издательств и технический директор венчурной фирмы.

В начале девяностых, когда Linux едва появился на свет, в нем сидели только некоторые помешанные на компьютерах студенты. И это не удивительно, ведь заставить его работать могла лишь черная магия, и, в отличие от коммерческих операционных систем того времени, Solaris и Windows, у Linux не было официальных каналов поддержки. Позже Linux стал более доступным простым пользователям, и в компаниях SuSE (1992) и Red Hat (1994) появились первые коммерческие службы поддержки. Эти фирмы поняли, что если что-то свободно, то это еще не значит, что нельзя зарабатывать деньги на создании коммерческих сервисов для этого.

Когда вы покупаете дистрибутив Linux сегодня, в вашем распоряжении оказывается «набор первой помощи» – обновления обычно загружаются автоматически, и у вас есть доступ к базе знаний в Интернете. В некоторых случаях вы получаете и техническую поддержку в течение определенного срока с возможностью продления контракта по его истечении. По просьбе *Linux Format* я проверил, как работают эти службы поддержки.

Как же оценить их работу? В принципе, просто: вы устанавливаете продукт, возможно, специально «роняете» его, а затем звоните в службу поддержки и притворяетесь новичком. Мы проделали это с Red Hat, Novell (производитель и поставщик SUSE) и Ubuntu; Red Hat и Novell были выбраны потому, что это самые известные производители коммерческих дистрибутивов Linux, а Ubuntu – потому, что недавнее появление службы поддержки Canonical представляет собой интересную альтернативу двум известным фирмам.

PR-службы каждой из этих компаний заранее знали о наших намерениях и, конечно, могли шепнуть службам поддержки, чтобы они ждали моих звонков. Полагаю, они этого не сделали – по крайней мере, кто-то из Canonical даже подозревал меня в том, что я пытаюсь получить техподдержку для двух компьютеров по договору для одного. Сотрудники Red Hat решили, что для подписания официального договора нет времени, и вместо этого мне дали прямой электронный адрес суппорта. К счастью, примерно в середине процесса тестирования для меня была создана стандартная учетная запись, и я мог отправлять отчеты как обыкновенный пользователь без особых привилегий!

Суета вокруг дивана

Сотрудникам служб поддержки я сообщал, что у меня небольшая фирма в несколько десятков человек, и мы решили попробовать Linux. Я запросил копию продукта у каждого из поставщиков, предпочтительно в версии для рабочей станции, и договор на техническую поддержку. Моим тестовым компьютером стал Pentium III 600 МГц с 512 МБ оперативной памяти и парой сетевых карт NetGear FA311 – старенький, но вполне обычный настольный ПК. Каждый дистрибутив Linux устанавливался на отдельный жесткий диск. И хотя такая установка выглядит несколько устаревшей, с этой проверенной временем конфигурацией компьютера я был уверен в том, что не столкнусь с проблемами из-за несовместимости дистрибутива и аппаратного обеспечения.

Я задал сотрудникам служб поддержки несколько вопросов:

» Проблема 1: Жесткий диск быстро переполняется.

Такая проблема появлялась у меня чуть раньше на сервере баз данных *MySQL*. Тот малый, который его настраивал, установил небольшой скрипт. Этот скрипт раз в несколько секунд проверял, запущен ли *MySQL*. Если *MySQL* не был запущен, скрипт отправлял письмо системному администратору. И когда мы на время остановили сервер *MySQL* чтобы обновить его, то совсем забыли об этом скрипте, и он стал перегружать процессор, посылая тонны писем. Одновременно разрастался и журнальный файл почтовика. Для имитации этой ситуации на тестовом компьютере был запущен скрипт, который с периодичностью в секунду или две отправлял письмо одному из пользователей.

Решение: Из полученной от меня информации, сотрудники службы поддержки по крайней мере должны определить, что сервер электронной почты перегружает процессор, и его журнальные файлы переполняют диск.

» Проблема 2: Я уверен, что правильно настроил общий доступ к файлам, но пользователи рабочих станций Windows не могут войти в систему.

Очень часто нужно обеспечить доступ к файлам Linux с компьютеров, на которых установлена Windows. Для этого используется пакет *Samba*, который позволяет пользователям Windows получить доступ к файлам на компьютерах с Linux. В некоторых дистрибутивах необходимо добавить к существующим в системе учетным записям дополнительные, которые будут использоваться *Samba*. Если этого не сделать (а я этого не сделал), пользователи не смогут войти в систему.

Решение: Необходимо добавить пользователя(-лей) в файл паролей *Samba* с помощью команды **smbpasswd -a**.

» Проблема 3: Сеть на второй сетевой карте не работает.

Конечно, чтобы решить другие проблемы, нужно было подключиться к сети, но я решил немного позабавиться с неработающей сетью. Недавно у одного из моих коллег были непонятные проблемы с сетью. Дело было в сетевой карте, некоторые элементы на которой сгорели. Устройство распознавалось системой, но сеть не работала. Я имитировал эту ситуацию с помощью неисправной сетевой карты, хотя ожидать от службы техподдержки прямого ответа «кто-то взял самые главные провода и закоротил их» было бы излишне оптимистично.

Решение: Наверное, ваша сетевая карта неисправна.

» Проблема 4: Могу ли я изменить размер раздела подкачки?

Эта одна из тех проблем, с которыми я сталкиваюсь постоянно. Кто-то во время установки делает что-то неправильно, например, выбирает размер раздела подкачки слишком маленьким. Или, что более вероятно, вы хотите установить коммерческую СУБД, а ей необходим огромный раздел подкачки. На моем сервере размер раздела подкачки был задан установщиком по умолчанию. Я спросил ребят, можно ли его изменить.

Решение: Простого способа изменения размера раздела подкачки «на лету» нет. Однако вы можете создать файл подкачки или добавить еще один диск и создать на нем дополнительный раздел.

Посмотрим, смогут ли службы поддержки помочь мне починить тщательно смоделированные неисправности в моей фиктивной системе. Все три фирмы позволяют связаться с ними по телефону, отправить письмо по электронной почте или послать описание проблемы с их сайта в Интернете. В каждом случае я старался опробовать все варианты.

Red Hat

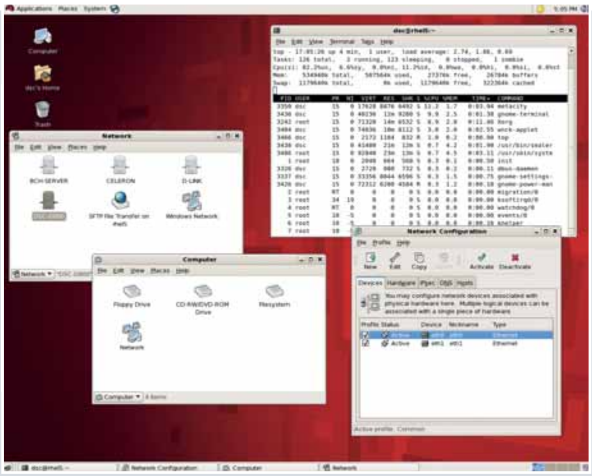
ОПИСАНИЕ ПРОДУКТА

Тестируемый продукт: Red Hat Enterprise Linux 5.0

Тестируемый сервис поддержки: 90-дневная поддержка в режиме 24x7 по программе 'structured evaluation'

Стоимость: $339 без НДС за год телефонной поддержки с 9 до 9 (в США) или с 9 до 5 часов (международная) в стандартном варианте. Доступны также базовые варианты. Для серверных продуктов доступна поддержка 24x7.

Web-сайт: www.redhat.com/rhel



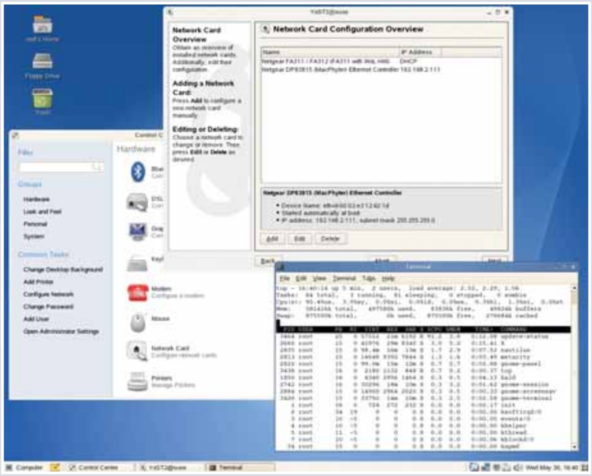
Я отправил сообщения о проблемах 1, 3 и 4 по электронной почте (кратчайшим путем, потому что пользователя для официальной службы поддержки мне еще не завели). Прошло чуть больше часа, потом я получил ответ, в котором они спрашивали номер телефона, по которому со мной можно связаться. Несколько минут спустя зазвонил телефон. Технарь на том конце провода, несомненно, был очень умен и с ходу решил проблему 4 (на самом деле он предложил кучу вариантов, о которых я даже не подозревал, причем все они были полезными и осмысленными). Он также предложил оформить официальные заявки для остальных запросов и сказал, что уже создал учетную запись для меня в системе поддержки; позже через эту систему я отправил сообщение о второй неисправности.

Проблемы были решены просто изумительно. Первая, как они полагали, была вызвана перегрузкой почтового сервера, что приводило к разрастанию его журнальных файлов и переполнению жесткого диска. Что касается проблемы с сетевой картой, они предложили запустить некоторые диагностические средства, на основе выходных данных которых они могли бы проверить работу карты на низком уровне и/или определить, входит ли моя сетевая карта в «сертифицированный» список Red Hat. Со второй проблемой Леонардо, один из них, справился с первой попытки.

Для получения самого быстрого решения потребовалось час и двадцать минут, а для самого медленного – около 2:20. Ребята, с которыми я говорил по телефону, оказались очень дружелюбными и полезными, а на запросы, отправленные через сайт, поступили очень лаконичные и ясные ответы. Во многих случаях специалист мог попросить меня запустить некоторые команды или средства диагностики и отправить ему результаты, но даже когда это было непросто и запутанно, они давали мне четкие инструкции, и я не мог ошибиться.

Novell

ОПИСАНИЕ ПРОДУКТА
Тестируемый продукт: SUSE Linux Enterprise Desktop 10
Тестируемый сервис поддержки: Мы использовали существующий многоцелевой договор 24x7, но его ближайший эквивалент – это Novell Priority Support
Стоимость: $3513 (без НДС) за год поддержки по телефону 12x5 и поддержки остальных продуктов Novell включая SLES (Tech Select)
Web-сайт: www.novell.com/linux



Было не до конца понятно, как отправить запрос через сайт. В службе по работе с клиентами по e-mail мне посоветовали отправить электронное письмо. В течение часа никто не ответил, и я позвонил по телефону.

Сначала мне ответила сердитая дама, недовольная тем, что у меня не было подходящего идентификатора пользователя (идентификатор клиента службы поддержки отличается от кода активации продукта), но со второй попытки мне попалась настоящая американская женщина, которая смогла дать мне правильный идентификатор пользователя для службы поддержки по коду активации SUSE. Я сообщил ей подробности второй проблемы (*Samba*), и она сказала, чтобы остальные запросы я отправил через сайт службы поддержки.

Как и в Red Hat, в некоторых случаях меня попросили загрузить и запустить некоторые диагностические средства. По второй проблеме меня сначала попросили предоставить подробную информацию, и как только я это сделал, проблема была тотчас же решена. По первой проблеме я получил ряд предположений, одно мимо цели (неисправность системы настольного поиска) и одно в яблочко (журналы почтового сервера переполняют диск). По поводу проблемы с сетевой картой они посмотрели на диагностическую информацию, которую я им отправил, и сказали: «Вам нужно проверить, исправна ли эта карта. Прежде у Вас не было с ней проблем?» Прекрасно.

Единственное разочарование, которое ожидало меня с Novell, относилось к вопросу об изменении размеров раздела подкачки (проблема 4). Сначала мой запрос решили перенаправить кому-то в моем часовом поясе. В этом был бы смысл, если бы ее решение требовало большой активности с моей стороны. Но этот запрос был из серии «один вопрос – один ответ». В любом случае, время от отправки запроса до получения приемлемого ответа составило около двух часов. Кроме миссис Грампи [Grumpy – сердитый, раздражительный, – прим. перев.], все отвечавшие на том конце провода были очень вежливыми и квалифицированными, и хотя проблемы, похоже, решала группа технических специалистов, задания между ними распределялись отлично, и ответы поступали вовремя.

Canonical

ОПИСАНИЕ ПРОДУКТА
Тестируемый продукт: Ubuntu 7.04
Тестируемый сервис поддержки: Поддержка пользователей рабочих станций 24x7
Стоимость: $250 без НДС за год телефонной поддержки 9x5, $900 без НДС за год телефонной поддержки 24x7.
Web-сайт: www.ubuntu.com/support/paid



Поскольку Canonical дает возможность отправить запрос к службе поддержки через сайт, я решил так и сделать с запросами 1, 3 и 4. Через полчаса после отправки третьего запроса (проблема с сетевой картой), я получил электронное письмо, в котором сообщалось о наличии свежей информации по данному пункту. Специалист спрашивал меня: «Можете ли Вы поменять конфигурацию сетевых карт таким образом, чтобы интерфейс eth0 был настроен на статический IP, а eth1 – на DHCP? Возможно, что-то с сетевой картой, и если после этого DHCP-сеть перестанет работать, а со статической будет все в порядке, то карту нужно заменить.» Решено. Четвертая проблема также была полностью решена достаточно быстро (около часа). В первой проблеме они правильно предположили, что все дело в почтовом сервере – на самом деле мне даже очень дипломатично сказали, что они подозревают меня в попытке получить поддержку для двух серверов по договору обслуживания одной рабочей станции и поинтересовались, зачем я запускаю почтовый сервер в настольной версии продукта! Итак, пока все хорошо.

Чтобы решить проблему с *Samba*, я решил позвонить по телефону. Несколько раз я слышал сигнал «занято», но на третий раз услышал механический голос, который очень комично произносил «Ууууу-б-ууунтуууу». Потом я где-то минуту говорил с очень дружелюбным и знающим свое дело человеком. Он задал мне несколько важных вопросов, спросил мой идентификатор пользователя и попросил меня собрать несколько конфигурационных файлов. Мы условились, что как только я это сделаю, мы продолжим общение через web-сайт службы поддержки, и после интенсивного обмена вопросами и ответами мы подобрались к решению очень близко (команда, которую мне предложили выполнить, выдавала сообщение об ошибке, но в ее описании в разделе "SEE ALSO" была ссылка на *smbpasswd*, и следующий шаг был очевиден). Самое быстрое решение проблемы потребовало 30 минут, самое медленное – чуть более пяти часов. Важно понимать, что мы не бездействовали все это время. В большинстве случаев мы активно контактировали друг с другом: они задавали вопросы, а я предоставлял необходимую информацию.

Результаты

В приведенных ниже таблицах показаны обобщенные результаты для каждой из компаний. Число взаимодействий – это общее число звонков, сообщений электронной почты и посещений сайта как мной, так и специалистами службы поддержки, включая запросы и ответы, и исключая формальные сообщения типа «Спасибо, этот вопрос закрыт» и автоматические сообщения системы типа «Статус изменен с X на Y».

Red Hat

Проблема	Отправлена	Решена	Число взаимодействий	Время (ч)
1	По e-mail	На сайте	2	2:17
2	На сайте	На сайте	2	1:21
3	По e-mail	На сайте	2	1:43
4	По e-mail	По телефону	2	0:40

Novell

Проблема	Отправлена	Решена	Число взаимодействий	Время (ч)
1	По e-mail [1]	На сайте	2	Около 2:00 [2]
2	По телефону	На сайте	4	Около 2:00
3	По e-mail [1]	На сайте	4	Около 2:00
4	По e-mail [1]	Не решена	3	Около 2:00

Canonical

Проблема	Отправлена	Решена	Число взаимодействий	Время (ч)
1	На сайте	На сайте	4[3]	4:31
2	По телефону	На сайте	9	0:41
3	На сайте	На сайте	2	0:45
4	На сайте	На сайте	4[3]	3:15

[1] Перенаправлена на сайт, [2] Примерно, поскольку система формирования отчетов содержит только дату, но не время. [3] Это время и число взаимодействий, которые потребовались на получение ответа, который я хотел услышать. Но в обоих случаях были также дополнительные указания о том, что именно делать, и другие полезные советы.



Наш рейтинг служб поддержки

	Red Hat	Novell	Canonical
Помощь по телефону (макс. – 5 баллов)[1]	5	3	4
Online-поддержка (5 баллов)	4	3	5
Время решения проблем (5 баллов)	5	5	5
Правильность ответов (4 балла)	4	3	4
Итого (максимум – 19 баллов)	18	15	18

[1] В случае с Red Hat я не сделал ни одного запроса по телефону, но они перезвонили мне, чтобы я мог с ними поговорить.

Проблемы решены

Конечно, краткость – сестра таланта Мне не в чем упрекнуть службу поддержки Ubuntu (в лице Canonical) – они работали очень оперативно и с пользой. Даже в том единственном случае, когда они не смогли найти верное решение, они двигались в правильном направлении. И хотя, когда я звонил им, я часто слышал сигнал «занято» это все же лучше, чем сидеть в очереди и платить за место поближе к началу.

Служба поддержки Red Hat также была расторопной. Хотя они больше полагались на результаты работы средств диагностики, в этом нет ничего плохого – я бы предпочел, чтобы они запросили больше информации, чем гадали вслепую. Мне очень понравился специалист, с которым я говорил об изменении размеров раздела подкачки – он разбирался в этом очень хорошо.

Хотя я был немного разочарован, когда один парень из поддержки Novell передал мою проблему другому, это было только с одной проблемой из четырех. Они очень вежливо попросили меня предоставить некоторые низкоуровневые данные о работе сетевой карты, да и во всех остальных случаях они работали хорошо и находили решение проблемы очень быстро. Человек, с которым я говорил по телефону, был очень интересным и всегда задавал нужные вопросы.

Коль скоро между этими фирмами так сложно выбрать, рассмотрим системы, которые используют на своих сайтах службы поддержки. Если у вас большая корпорация с большим количеством рабочих станций Linux, то будут сложные вопросы, разрешить которые можно только по телефону. На рынке товаров массового спроса удобнее всего отправлять запросы через сайт. Даже если сначала вы связываетесь по телефону, в дальнейшем статус вашего запроса проще отслеживать через Интернет.

Canonical, похоже, внедрила систему собственного производства – или во всяком случае, коммерческую систему, авторы которой стремились не раздувать продукт сверх необходимого. К описанию проблемы можно приложить документы, а когда специалист отвечает на ваш вопрос, вам приходит оповещение по электронной почте – хотя немного неприятно то, что подробности ответа не включаются в письмо, и вам нужно зайти на сайт и прочитать его. Похоже, Red Hat использует *Oracle CRM*, которая (как и многие коммерческие системы такого типа) немного медленная из-за большого размера, но в то же время достаточно удобная, и в случае каких-то изменений вам также приходит оповещение по электронной почте. В отличие от Canonical, в это оповещение включается подробное описание всех изменений.

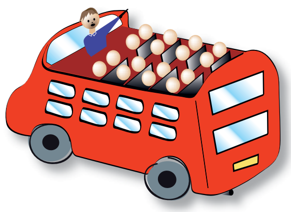
В Novell используется наименее удобная из всех трех систем: *Siebel eService*, которая не отправляет пользователю никаких оповещений. Система также не отличается удобством навигации, особенно когда нужно переключиться между различными запросами внутри вашей учетной записи.

В общем, я так и не смог разделить первое место между Red Hat и Canonical. Мой опыт показал, что в службах поддержки всех трех компаний работают компетентные сотрудники, которые задают осмысленные вопросы и решают поставленные задачи в приемлемое время. Общение с ними происходит через удобные системы. Может это выглядит как слишком вежливое заключение, но поверьте: вам нечего бояться, когда речь заходит о корпоративной поддержке Linux. Пользователи Linux как из государственного, так и из частного сектора могут чувствовать себя в надежных руках. LXF



Illustration by: Seth Singh

Большое Linux-турне по файловой системе



Заблудились в файловой системе Linux? Отчаялись понять, в чем тут смысл? **Д-р Крис Браун** прописывает вам путешествие по файловой системе.

«Привет! Добро пожаловать в путешествие по файловой системе Linux. Меня зовут Ман Страницын, и на сегодня я ваш гид. Я и мой водитель автобуса, Хал Д., рады видеть вас на борту. Пара предупреждений, прежде чем мы отправимся: пожалуйста, ни в коем случае не высовывайте руки из автобуса и не удаляйте ничего, что бы ни встретилось на пути, не будучи полностью уверены в том, что делаете. Итак, отправляемся.

Мы начинаем нашу экскурсию с корневого каталога, и, я полагаю, вы можете сказать, что отсюда, хм, много чего видно, так как это самый верхний уровень иерархии каталогов. (Дико извиняюсь, остроты сочинял не я: я лишь читаю вслух сценарий.) Любопытно, что корневой каталог, в отличие от прочих каталогов, безымянен. Вам могут сказать, что его имя – /, но это не так. Просто, записывая абсолютный путь, мы начинаем его с /, а в случае корневого каталога тут писать нечего.

Слева по ходу вы заметите каталог с именем **/root**. Не перепутайте: это не корневой каталог, а просто каталог с именем **root** [root – англ. корень, прим. пер.]. Вообще-то, это частная территория системного администратора, она же – домашний каталог суперпользователя. Мой дядя, системный администратор ОС Solaris, вечно ноет, что у него нет собственного домашнего каталога: его домашний каталог /, и он его ненавидит. А вам понравилось бы развешивать белье на главной

городской площади? Можем мы осмотреть **/root**, Хал? Ой, похоже, что нет – полковник Линукс говорит «нет». Что ж, за спрос денег не берут, но **/root** – один из немногих каталогов, просмотр которых обычным пользователям запрещен. Вообще говоря, политика доступа в Linux – «смотри, но не лапай». Единственное исключение – домашние каталоги пользователей: там они могут вытворять все, что угодно.

Кстати, о полковнике Линуксе – наша следующая остановка называется **/boot**, место, где он живет, так что приготовьте фотоаппараты или кнопки Print Screen. Здесь находится файл с именем, похожим на **vmlinuz-2.6.19**, это (сжатый) образ ядра Linux. При загрузке Linux, загрузчик (обычно *Grub*) помещает этот файл в память и запускает его. Вы также можете заметить файл с именем типа **initrd-2.6.19.img**, являющийся образом стартового RAM-диска. Он содержит модули ядра, необходимые во время загрузки, до того, как будет получен доступ к самой файловой системе. Не удаляйте эти файлы, ребята, не то вы не сможете загрузиться, а я стану безработным.

А, ребята на задней площадке заметили каталог с именем **lost+found**. Молодцы! Вы увидите этот каталог на верхнем уровне любого раздела, содержащего файловую систему ext2 или ext3. Зачем он? Вы, небось, подумали, что это место встречи для отбившихся от BSD, ха-ха, но он нужен программе под названием *fsck*, проверяющей целостность файловой системы. Если *fsck* находит целый на вид файл, не имеющий имени, то создает запись для него в **lost+found**. Если честно, теперь это случается крайне редко, так что, скорее всего, **lost+found** просто пуст. Можете без проблем оставить его в покое.

Город настроек

Ну вот, справа от вас вы можете видеть каталог, названный **etc**. Полагают, что исторически это имя произошло от 'et cetera' (в переводе, «и прочее»): туда помещалось все, что нигде больше не удалось пристроить. Мы закидывали в него что попало, но теперь он стал пристанищем большого числа конфигурационных файлов системы и скриптов. Некоторые из этих файлов критичны; например, **/etc/passwd** содержит информацию обо всех локальных учетных записях (включая root). Вы также можете заметить здесь **/etc/inittab**, который сообщает *init*, что делать (*init* – очень важная программа: при загрузке Linux только она одна автоматически выполняется ядром и отвечает за запуск всех остальных служб, включая службы входа).

Не менее важен и **/etc/fstab**, говорящий о том, какие еще файловые системы должны быть смонтированы. Вам лучше не шутить с ними, если вы не знаете что делаете; ошибки в этих файлах могут привести к невозможности загрузки или входа в систему. В **/etc** вы найдете файлы конфигурации для различных служб сети – это **/etc/xinetd.conf**, содержащий настройки *xinetd*, и **/etc/syslog.conf**, настраивающий *syslog*. Все эти файлы, кстати, хранятся в обычном текстовом формате, и именно поэтому единственный инструмент, необходимый для настройки Linux – это текстовый редактор. Мой любимый редактор *Vi*, но Хал говорит, что я и вообще по жизни чудик.

Поскольку **/etc** справа мы осмотрели, то посмотрим на **/home** слева. Это наш жилой район. Внутри него находятся дома-каталоги отдельных пользователей. Например, домашний каталог Хала – **/home/hal** (правда, Хал?). Население тут весьма умное, но все зависит от соседей. Некоторые валят все в одну директорию, другие организуют для своих файлов многоуровневые каталоги с говорящими именами. По умолчанию, внутри **/home** даются права просматривать, но не изменять, файлы других пользователей. Естественно, пользователи могут и запретить просмотр своих файлов. У юного Тома есть каталог с именем **/home/tom/photos**, доступ к которому имеет только он. Том, мы спим и видим узнать, что ты там прячешь...

Каталог **/mnt** Хал хочет проскочить побыстрее. Боюсь, ребята, что здесь интересного мало – лишь пустые каталоги для временно монтируемых файловых систем. Сразу после **/mnt** мы перемещаемся в **/media**, он содержит подкаталоги, используемые как точки монтирования для съемных устройств типа дискет и CD-ROM. Молодежь, кто помнит, что такое дискета? Короче, идея в том, что система горячего подключения на лету автоматически монтирует носители сюда, стоит только их вставить.

Мы исключили осмотр этого места из экскурсии после происшествия в прошлом месяце, когда Хал повел автобус в **/media/cdrom** просто чтобы доказать, что он пуст. Затем кто-то вставил CD, демон горячего подключения проснулся, и вдруг перед нами появилась иерархия диска с отпускными фотографиями. Это, скажу я вам, был полный шок.

Виртуальная часть экскурсии

Так, каталог, в который мы направляемся, **/proc**, действительно интересен, потому что на самом деле не существует. Все файлы в нем – игра воображения ядра: они не соответствуют никакой информации, реально расположенной на диске. Полковник Линукс на прошлой неделе объяснил мне, что эти внутренние структуры данных помогают следить за такими вещами, как использование памяти, и хранить информацию »

Про FHS

На сайте www.pathname.com/fhs есть документ под названием Filesystem Hierarchy Standard (FHS) [Стандарт иерархии файловой системы]. Это не официальный документ стандартов (он слишком короток, слишком неформален и слишком легко читается, чтобы под ним подписались ANSI, IEC или ISO!), но это наиболее полное описание того, где что должно располагаться в файловой системе Unix/Linux и почему.

Флора и фауна файловых систем

Имеется семь типов «существ», живущих в файловой системе. Когда вы выводите длинный список каталогов (**ls -l**), первый символ каждой строки говорит о типе данного элемента. Они показаны в первом столбце таблицы. Мы также выполнили подсчет популяции каждого типа в системе Ubuntu; эти значения приведены во втором столбце. Понятно, что, как выражаются продавцы автомобилей, «расход бензина на километр имеет индивидуальные отличия».

Тип	Численность	Описание
-	102,314	Обычный файл. Наиболее распространенный тип.
d	14,701	Каталог. Каталог – это контейнер для других элементов. Некоторые называют их папками.
l	15,258	Символьная ссылка. Это крошечные файлы, содержащие имя какого-то другого файла, вроде ярлыков в Windows. Поэтому, если, например, у меня есть символьная ссылка **/etc/motd на /var/run/motd**, и программа открывает **/etc/motd**, то ядро говорит: «Ага, это символьная ссылка. Он хотел сказать не **/etc/motd**, а **/var/run/motd**», и открывает последний. Символьные ссылки иногда называют «симлинками» (symlinks) или гибкими ссылками.
c	785	Так называемое символьное устройство (иногда также называемое raw-устройством или специальным символьным файлом). Этот элемент служит для именования устройств. Нечто вроде **/dev/console** соответствует реальному физическому устройству. Другие соответствуют псевдоустройствам; например, **/dev/random** предоставляет доступ к генератору случайных чисел ядра.
b	65	Блочное устройство. Блочные устройства – это обычно диски: скажем, **/dev/hda2** – это второй раздел первого жесткого диска IDE. Вообще говоря, символьные устройства поддерживают чтение и запись последовательного потока байтов, а блочные устройства поддерживают произвольный доступ. Реальное отличие, однако, заключается в том, что ядро предоставляет блочным устройствам буферный слой, чтобы они могли читать и писать целыми блоками. Для символьных устройств этого не делается. Практически все файлы устройств обитают в **/dev**.
s	34	Сокеты Unix-домена. Они называются «конечными точками связи в файловой системе» и используются почти так же, как TCP- и UDP-сокеты, за исключением того, что они поддерживают лишь межпроцессное взаимодействие для запущенных процессов на той же машине.
p	7	Именованный канал. Они так редки, что их следует занести в красную книгу! Подобно сокетам Unix-доменов, это именованные конечные точки, используемые для межпроцессного взаимодействия.



» о каждом процессе, типа переменных окружения. В старые времена команды вроде *ps* (отображающая информацию о запущенных процессах) вылавливали свою добычу в образе памяти ядра. Полковника это в восторг не приводило: он говорит, что это смахивало на вскрытие живого пациента. Поэтому он пришел к мысли сделать эту информацию доступной в виде набора файлов. Таким образом, программа может разжиться необходимой информацией, просто открыв и прочитав эти воображаемые файлы – так же, как любой другой файл.

Хал заедет в **/proc**, и мы там оглядимся. Вы можете вообразить, что с **/proc** что-то неладно: ведь если выполнить здесь *ls -l*, то большинство файлов будет иметь нулевой размер; но если просмотреть их содержимое при помощи *cat* или *less*, то они не пусты!

Я уже говорил, что так называемые местные файлы показывают нам содержимое внутренних структур данных ядра в виде простого текста. Например, файл **cmdline** покажет нам параметры, с которыми загрузилось ядро. Файл **cpuinfo** покажет, что знает ядро о центральном процессоре (или процессорах), на котором оно запущено. Файл **meminfo** расскажет нам о системе виртуальной памяти столько, сколько нам и знать-то не надо. И так далее.

Пардон, мадам, не сядете ли вы на место? Хал собирается выполнить крутой разворот, чтобы показать вам коллекцию каталогов с именами типа **3412**. Эти имена соответствуют ID процессов, а их каталоги содержат еще больше виртуальных файлов, которые обеспечивают доступ к информации о каждом процессе. Вообще-то по ним имеется документация (для справки наберите **man 5 proc**), но большая часть информации столь низкого уровня, что будет малопонятна среднему туристу. В большинстве случаев лучше использовать программы типа *top* и *ps*, которые покажут вам информацию о процессах в более удобной форме.



Урок истории

Имя 'tty' когда-то означало 'teletype'. Телетайп – это механическое печатающее устройство с клавиатурой типа печатной машинки. В частности, одна модель, ASR33, была чрезвычайно популярна на мини-компьютерах в 1970-х, когда даже полковник Юникс был еще лейтенантом, а полковник Линукс был лишь искоркой в глазах своего родителя. Телетайп давно исчез, а имя прилипло. Сейчас tty – символьный экран любого типа.

По большей части, мы воспринимаем **/proc** как файловую систему только для чтения, но на самом деле в **/proc/sys** имеется несколько «файлов», содержащих различные параметры подстройки ядра, которые вы можете изменить, просто вписав их туда. Например, можно уменьшить параметр под названием **TCP FIN TIMEOUT** с 60 до 50 следующим образом:

```
# cd /proc/sys/inet/ipv4
# cat tcp_fin_timeout
60
# echo 50 > tcp_fin_timeout
# cat tcp_fin_timeout
50
```

Ну-ка, поднимите руки те, кто понятия не имеет, что такое **TCP FIN TIMEOUT** и зачем его менять. Вижу: большинство. Так я и думал. Для 99% из нас лучше всего даже не притрагиваться к нему.

От **/proc** мы спустимся по улице к **/sys**. Это еще одна из «виртуальных» файловых систем. Она была добавлена в ядро 2.6, чтобы упростить коду уровня ядра, например, драйверам устройств, обмен данными с программами пространства пользователя. Иерархия **/sys** позволяет вам увидеть состояние устройств (шины, устройства и так далее), обнаруженных ядром, и если только вы не переписываете, скажем, подсистему горячего подключения, советую полностью ее проигнорировать. Вся документация есть в книге *"/proc et /sys"*, написанной Оливье Доделем [Olivier Daudel] и опубликованной издательством O'Reilly... по-французски.

Остановка по требованию

Теперь мы собираемся посетить **/dev**. «Dev» – это сокращение от «devices» [устройства], и здесь живут странные создания. Они ведут себя не как обычные файлы. Если вы выполните здесь *ls -l* и внимательно посмотрите, то увидите в строке первый символ 'b' или 'c'. 'b' – это так называемые блочные устройства, то есть устройства с блочной структурой и произвольным доступом – обычно это разделы жесткого диска. Например, вот этот малютка, **/dev/hda1** – первый раздел жесткого диска, он является блочным устройством. В данной системе **hda1** – корневой раздел (на вашей машине может быть по-другому, в зависимости от того, как вы устанавливали систему); собственно говоря, все что мы посетили до этого, расположено в разделе, представленном этим малышом, так что он сильно занят. Как только вы нажмете Выключить компьютер, он отправится в ванну – затем какао, и спать.

Слева вы можете видеть большое скопление так называемых tty-устройств (привет, ребята!). Это символьные устройства, представляющие собой символьные терминалы. Например, Linux обычно настроен на поддержку шести виртуальных терминалов. Из вашего графического рабочего стола вы можете перейти к ним при помощи комбинации клавиш от Ctrl+Alt+F1 до Ctrl+Alt+F6 (и вернуться на рабочий стол при помощи Ctrl+Alt+F7). Все шесть виртуальных терминалов являются устройствами, от **/dev/tty1** до **/dev/tty6**.

Двигаемся дальше: прямо по курсу, в этой мрачной пещере, живет очень странный парень по имени **/dev/null**. Подобравшись поближе, мы сможем увидеть надпись над входом: «Оставь надежду всяк сюда входящий». Мы видели тех, кто входил сюда, но никто не вернулся обратно. Некоторые называют его черной дырой и используют для выбрасывания в него ненужного вывода программ. Э, кажется, мы слишком близко подъехали ко входу, Хал. Не въезжай туда… ты никогда уже... не въезжай туда, Хал! Заворачивай!

Фу, чуть не попали...

Городские задворки

Что ж, после всех этих виртуальностей вы, я уверен, хотите вновь увидеть какие-нибудь реальные файлы, и их много в том месте, которое я называю промзоной, состоящем из двух каталогов **/bin** и **/sbin**. Под bin подразумевается «binary» [двоичный]: большинство здешних файлов являются исполняемыми. Почему каталогов два? Ну, смысл в том, что



Изучаем корневой раздел

Одним из принципов организации файловой системы является возможность ее рационального размещения на нескольких разделах диска (или нескольких дисках), и возможность совместного использования ее частей несколькими машинами.

Ключевым здесь является понятие корневого раздела. Во время загрузки Linux ядро подключает к себе отдельный раздел файловой системы – он и называется «корневым». Любые другие разделы, которые необходимо подключить, монтируются командой *mount*, обычно в соответствии с записями в файле **/etc/fstab**. Поскольку на ранних стадиях загрузки доступна только корневая файловая система, она должна содержать все необходимое для функционирования Linux и присоединения других частей файловой системы. В инструменты корневого раздела входят: программа *init* (запускающая все остальные процессы), оболочка, *mount* и файл **/etc/fstab**. Стандарт иерархии файловых систем указывает количество каталогов, которые должны находиться в корневом разделе.

запускаемое обычным пользователем, типа *Vi* и *tar*, *rm* и *date*, помещается в **/bin**, а то, что может запустить только суперпользователь-администратор, расположено в **/sbin**. Можете считать, что «s» в **/sbin** происходит от 'system' – или, возможно, 'super-user'. Например, *ifconfig* (установка параметров сетевой карты) и *iptables* (определение правил брандмауэра) находятся в **/sbin**, потому что только суперпользователь имеет на них право. В большинстве дистрибутивов Linux **/bin** включается в пути поиска для обычного пользователя, а **/sbin** нет. Ясное дело, обычный пользователь может легко получить доступ к этим командам, набрав полный путь типа **/sbin/iptables**, но нормальной работы у него не выйдет, поскольку большинство из них и пальцем не шевельнут, если вы не супер!

Важный уголок нашей промышленной зоны называется **/lib**. Вообще-то это целый угол. «Lib» происходит от «library» [библиотека], и здешние файлы являются совместно используемыми библиотеками, необходимыми системным программам в **/bin** и **/sbin**. (Если вы пришли из мира Windows, то знаете их как DLL.) Критически важная библиотека, которая используется практически везде – это стандартная библиотека C, **libc.so**. Если вы удалите этот файл, то почти все остановится.

> **«Вы можете ощутить легкий толчок при переходе к /usr.»**

Осмотром промзоны экскурсия завершается; предлагаю по-быстрому заглянуть в очень важный каталог **/usr**. Я должен предупредить вас, что обычно **/usr** находится на другом разделе, так что вы можете ощутить легкий толчок при пересечении точки монтирования. Хал, сбрось, пожалуйста, газ!

Оглядевшись в **/usr**, мы увидим каталоги, на вид дублирующие некоторые из корневого раздела: в частности, **/usr/bin**, **/usr/sbin** и **/usr/lib**. И верно, они содержат элементы того же типа, которые мы уже видели в **/bin**, **/sbin** и **/lib**: а именно, **/usr/bin** содержит пользовательские команды, **/usr/sbin** содержит команды системного администратора, а **/usr/lib** – библиотеки. Так почему же эти элементы рассованы по двум различным наборам каталогов? Ответ – в действиях по выделению разделов. Элементы **/usr** часто расположены на отдельном разделе, который не подключается к файловой системе практически до последних стадий процесса загрузки. Действительно критичные компоненты, используемые на ранних стадиях загрузки, должны располагаться в корневом разделе, а не в **/usr**.

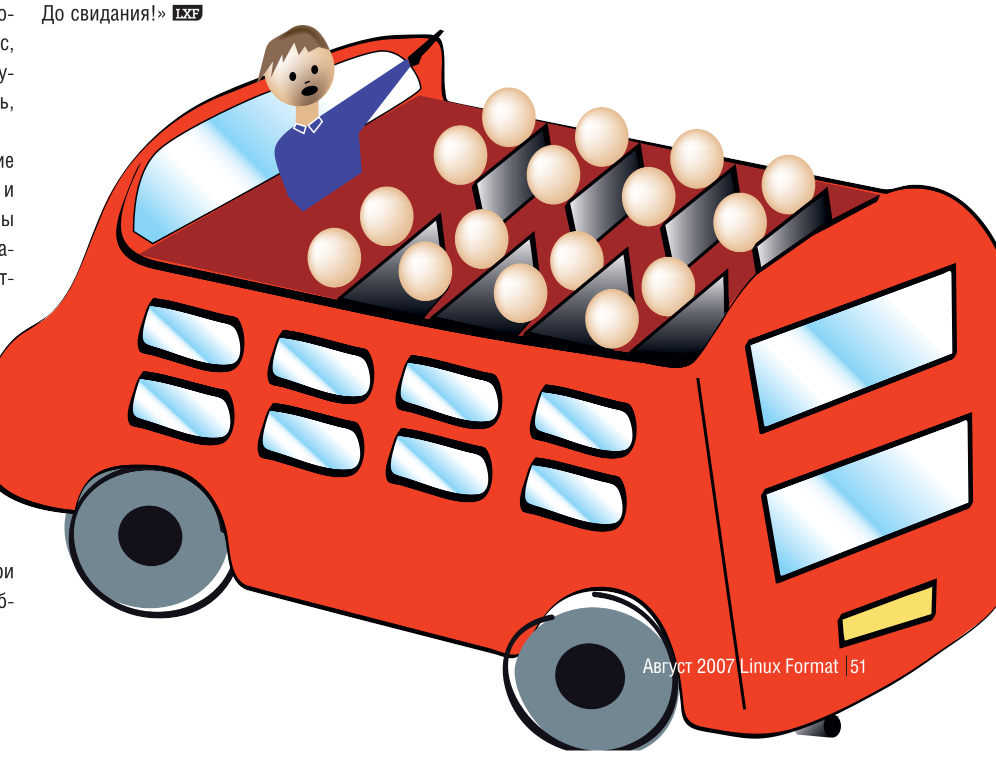
Распределение элементов таким способом также минимизирует те части файловой системы, которые не должны затрагиваться при однопользовательской загрузке. Элементы из **/bin**, **/sbin** и **/lib** необходимы, а из **/usr** – нет. Можно также смонтировать **/usr** в файловую систему в режиме «только для чтения», для большей безопасности, а в сети совместно использовать **/usr** с одного файл-сервера, по крайней мере для машин с общей аппаратной архитектурой.

Фактически большая часть исполняемых файлов и библиотек размещается в **/usr**, и, соответственно, малая часть в корневом разделе. Использование дискового пространства системой на машине, на которой я нахожусь в данное время, выглядит так:

```
$ du -sh /bin /sbin /lib /usr/bin /usr/sbin /usr/lib
 4.9M /bin
 6.3M /sbin
 109M /lib
 92M /usr/bin
 5.7M /usr/sbin
 625M /usr/lib
```

Цифры, которые вы увидите в своей системе, конечно же, будут другими, но смысл будет тем же: большая часть элементов расположена в **/usr**, а корневой раздел сравнительно невелик.

Есть еще несколько каталогов вроде **/var**, **/tmp** и **/opt**, которые мы не посетили, но я должен вернуть вас вовремя, чтоб вы успели ухватить несколько больших загрузок с FTP-зеркала, так что мы возвращаемся в корневой каталог, откуда начали, и завершаем нашу экскурсию. Будьте осторожны, выходя из автобуса. Надеемся, вы проведете несколько минут в нашей сувенирной лавке, где сможете купить брелки для открытых ключей с пластмассовым Туксом и открытки, говорящие «Я выполнил ls -R в **/proc** и выжил!». До свидания!» LXF



Что за штука... JavaFX?

Пол Хадсон проливает свет на новое зелье от Sun, соперника интерактивных web-приложений Flash, Ajax и Silverlight…

В позапрошлом номере на этих страницах был D. Месяц назад – E. Думаете, мне охота слушать о языках программирования три месяца подряд? Я уж лучше пойду…
Нет, нет – погодите! Не захлопывайте страницу!

Что?
Никакого программирования.

Ладно, у вас пять минут для того, чтобы заинтересовать меня. Время пошло.
Хорошо: JavaFX – новая крутая платформа от Sun для создания полноценных Интернет-приложений, конкурент Ajax. Это, кстати, та самая приправа, которой Google Mail обязан интеллектом. Ajax базируется на JavaScript, и это порождает целый спектр проблем совместимости с браузерами. Например, Google Docs прекрасно обрабатываются *Firefox*, но отказываются работать на *Safari*. Если вы пользователь Opera, то, как я слышал, *OpenOffice.org* хорош для редактирования документов. Но даже на *Firefox* Ajax не станет работать ни минуты, пока вы в оффлайне.

Значит, JavaFX призван заменить Ajax?
Я еще не закончил! Наряду с Ajax, для строительства привлекательных интерфейсов web-приложений все шире применяется технология Flash от Adobe, но, хотя он и работает оффлайн, *Flash Player* – все-таки не открытое ПО. Возникают проблемы, если у вас неподдерживаемое оборудование, или вам некогда ждать, пока Adobe наконец выпустит новую версию для Linux. И, конечно, если вы идейный противник проприетарного ПО, Flash тоже не для вас.

Подождите – так JavaFX заменит и Ajax, и Flash? Как бы не откусить больше, чем можешь прожевать…
Вот тут в дело вступает часть 'Java' от 'JavaFX': тут все основано на Java-платформе, то есть вся функциональность уже готова – Sun лишь надстраивает немного сверху. Например, есть Java 2D – библиотека, задуманная для высокоскоростной отрисовки графики, и она превосходна для создания блестящих графических интерфейсов. JavaFX располагает также всеми сетевыми и XML-библиотеками, необходимыми для отправки и получения данных по проводам, а также их кэширования оффлайн. Фактически, онлайн-приложения работают и в оффлайн-режиме.

Значит, экономится трафик?
Ну, объем передаваемого кода прежний, а браузер кэширует столько JavaScript, сколько можно, так что особой экономии ждать не приходится. Но JavaFX имеет определенные преимущества в вопросах безопасности, так как весь код исполняется внутри сверхбезопасной «песочницы» Java. Даже если у самого браузера проблемы с безопасностью, Java сохранит данные в неприкосновенности.

> «JavaFX-приложения будут работать на любом Java-совместимом устройстве. Да-да, и на вашем мобильнике тоже.»

Значит, JavaFX работает оффлайн, поддерживается многими браузерами, да еще и безопаснее. Мне начинает нравиться.
Я ж говорил, это здорово! На самом деле, предлагается много больше. Помните старый слоган Java: «Написано однажды, работает везде»?

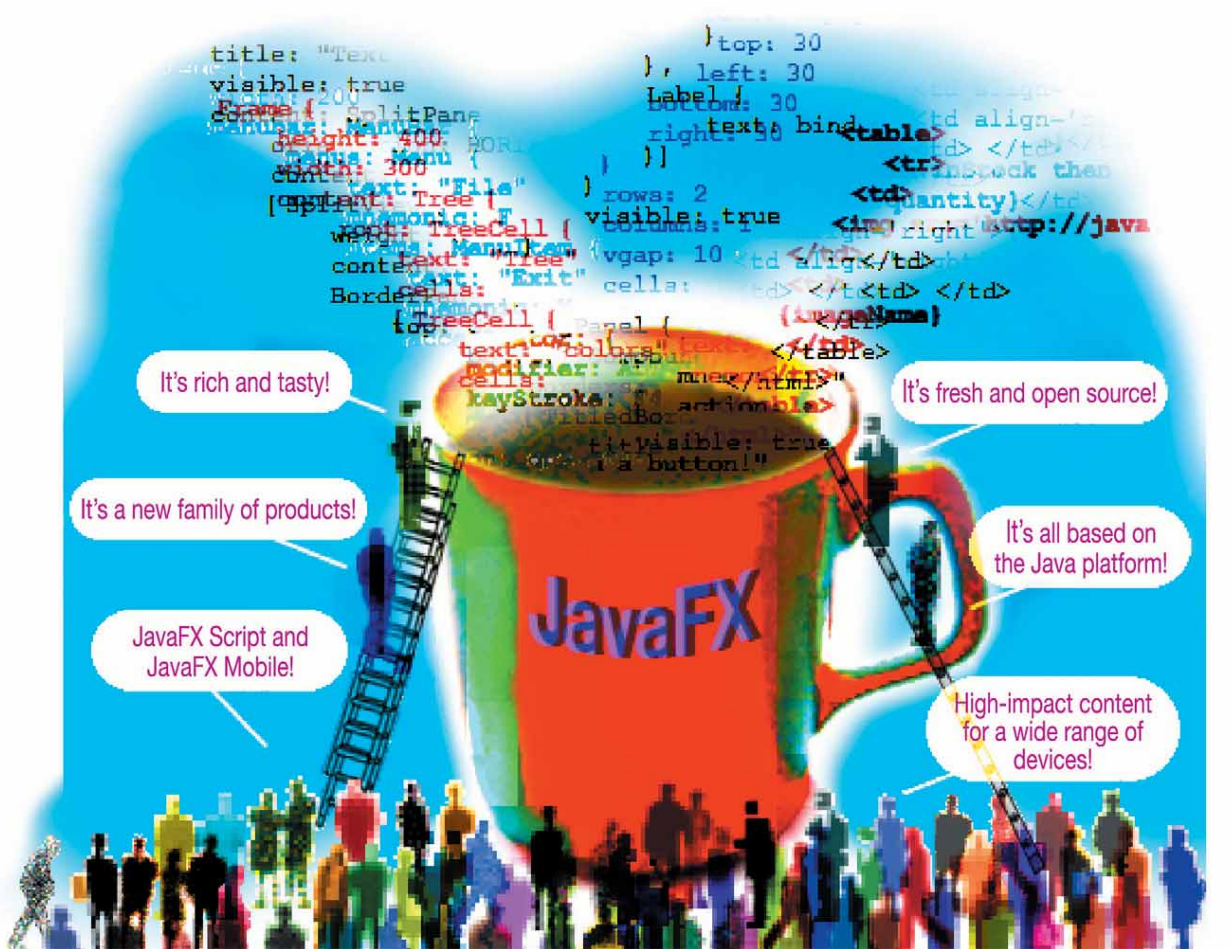
Мне помнится, было немного иначе: «Написано однажды, ломается везде».
Не спорю, раньше так и было. Но ведь Java с тех пор здорово усовершенствована, а это значит, что JavaFX будет работать на любом Java-совместимом устройстве. Да-да, и на вашем мобильном телефоне тоже. JavaScript очень медленно обрабатывается, потому и Google Doc на мобильных устройствах с их слабыми процессорами идут со скрипом, да и то если у браузера есть все необходимые средства для обработки JavaScript!

Но ведь и Java не из скороходов…
Напротив – язык Java для телефонов и других мобильных устройств существенно оптимизирован с целью извлечь максимум возможного из весьма ограниченных ресурсов. На телефонах не только слабые процессоры, но и весьма ограниченные объемы оперативной памяти, поэтому оптимизация Java пришлась как нельзя кстати – не нужны никакие хаки и прочая заумь, с помощью которых Ajax-приложения пытаются заставить работать на настольных ПК: все-таки Java – хорошо изученная платформа. Слой JavaFX просто ставится поверх виртуальной Java-машины (как на настольном ПК, так и на мобильнике), а это означает, что любой компьютер, инсталлировавший тонкий слой JavaFX, способен пользоваться подлинно кроссплатформенными JavaFX-приложениями.

А не выйдет ли так, что использование Java затруднит изучение JavaFX рядовыми программистами?
Это вполне могло случиться, если бы Sun одновременно не представила JavaFX Script.

Это что-то вроде JavaScript?
Не забывайте о том, что JavaScript – всего лишь претендент на трон Java. JavaScript не имеет ничего общего с Java, он даже изобретен не в Sun! JavaFX Script – который, будучи ядром JavaFX, вполне может называться просто «JavaFX» – тоже совершенно не похож на Java. JavaFX Script предназначен для упрощенного создания пользовательских интерфейсов с Java-библиотеками Swing, но он обладает полным доступом к основным библиотекам Java, если в том возникает потребность.

Мне смутно вспоминается что-то из прошлых лет. Grove? Groove?
Верно. Вы вспомнили Groovy, мы об этом писали в LXF67. Конечно, между Groovy и JavaFX есть определенное сходство, но JavaFX предназначен для создания web-приложений, и инструментарий у него соответствующий. Groovy был более общим, то есть недостающую функциональность нужно было достраивать самому.



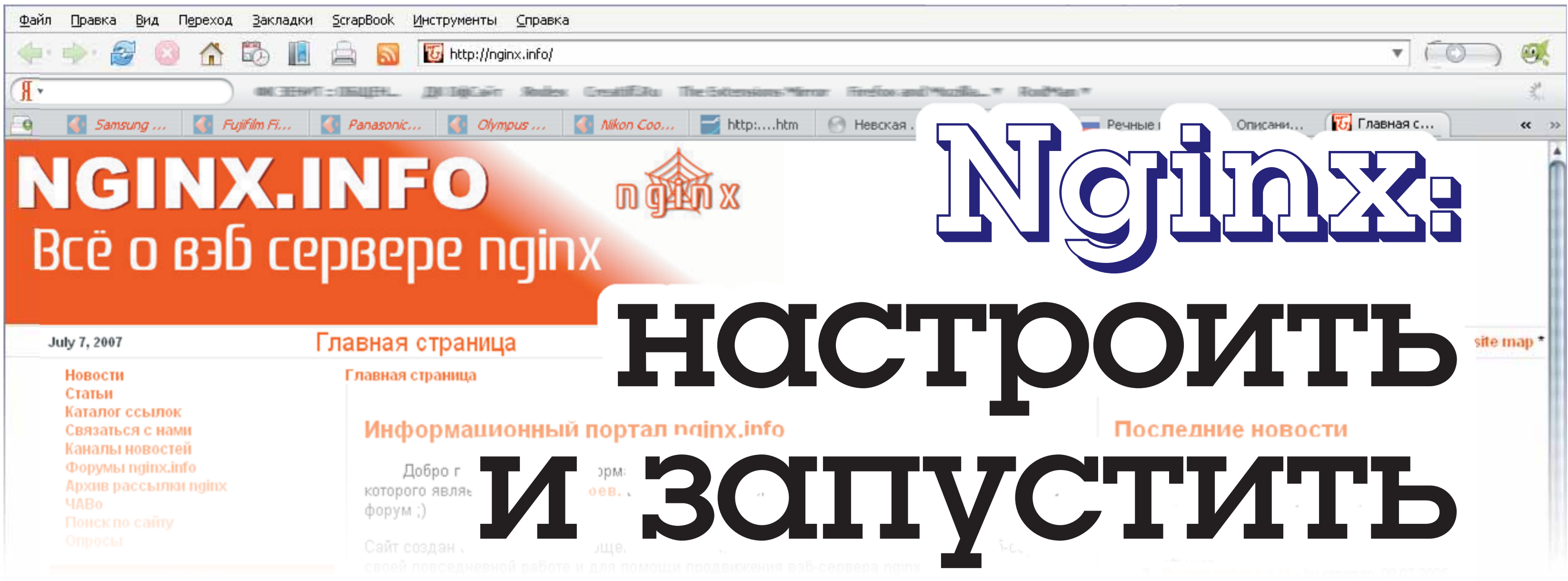
Понимаю, как это нудно. Как же Sun умудрилась так быстро все обстряпать?
На самом деле проект разрабатывался довольно долго, под условным названием F3 (Form Follows Function). Недавно Microsoft представила нечто очень похожее на JavaFX, под названием Silverlight. В Microsoft-варианте в web-браузер встраивается .NET вместе с графическим холстом, поэтому можно писать код на C# или JavaScript и получать примерно те же вещи, что с JavaFX.

Подождите, зачем тогда JavaFX?
Видите ли, несмотря на все наши надежды, Microsoft не сделала ощутимого поворота в своей политике по открытому ПО, а Sun уже открыла общедоступный сайт, на котором выложила GPL-лицензированный код JavaFX. Разработчики Mono тоже ухватились за идею Silverlight и начали работу над свободным клоном на основе Mono и Cairo. Но все это – дело будущего, а JavaFX есть уже здесь и сейчас, хотя и в альфа-версии, и образцы его кода уже можно «потрогать руками».

То есть, JavaFX можно попробовать прямо сейчас?
Конечно! Sun выпустила даже JavaFX Mobile, встраиваемый в мобильные операционные платформы на основе Linux. На выходе и другие продукты семейства JavaFX, но в наличии пока только JavaFX Mobile и JavaFX Script. В настоящее время код довольно сырой, поэтому не ждите от него чудес – но с помощью JavaFX уже можно строить и связывать между собой интерфейсы Swing.

Советуете немного подождать?
Наоборот, я рекомендовал бы включиться прямо сейчас, ведь первозданное состояние JavaFX позволяет вылепить из него все, что угодно, и на этой стадии ваши подсказки и пожелания очень бы пригодились.

Вы убедили меня как минимум загрузить и попробовать JavaFX. Назовите, пожалуйста, несколько URL, пока я в теме...
Вот они: домашняя страница JavaFX www.sun.com/software/javafx, там вы найдете ссылки на информацию, пресс-релизы и прочие маркетинговые детали. Если вы действительно решили ознакомиться с образцами кода, демо и руководствами, то вам сюда: http://openjfx.dev.java.net, это домашняя страница кода. Полезно посетить и такую страничку: http://blogs.sun.com/chrisoliver – блог того парня, который создал F3 и довел его до JavaFX. Там находится немало примеров кода, готовых к работе. Удачи вам! LXF



Nginx: настроить и запустить

ЧАСТЬ 2 Довольно теории – настало время пустить *Nginx* в дело. **Валерия Комиссарова** подскажет, как собрать его и настроить для работы в типовых ситуациях.



Наш эксперт

Валерия Комиссарова
Эксперт Cnews, дипломированный администратор и разработчик решений на C#, Валерия с интересом следит за развитием Unix и умеет по достоинству оценить существующие для него решения.

В прошлый раз мы познакомились с интересной разработкой под названием *Nginx*. Продукт создан нашим соотечественником Игорем Сысоевым и представляет собой web-сервер, а также HTTP- и почтовый прокси-сервер. Мы детально рассмотрели особенности *Nginx*, его функциональность и принципы действия. Необходимая предварительная информация получена – перейдем к обсуждению процесса работы с *Nginx*: установки, настройки и использования.

Nginx: инсталляция

Официальный сайт *Nginx* расположен на домашней странице его создателя, г-на Сысоева – http://sysoev.ru/nginx, где всегда можно найти последнюю версию продукта, небольшой комплект документации, список произведенных изменений (http://sysoev.ru/nginx/changes.html) и некоторую другую информацию. Текущая версия *Nginx* – 0.6.2 – находится в разделе «Скачать».

Nginx работает только под UNIX-подобными системами (разновидности Linux, BSD, Solaris и др.), поэтому архив с дистрибутивом продукта запакован традиционным для них образом и имеет расширение .tar.gz. Доступны также пакеты RPM/DEB, ebuild'ы Gentoo и порты FreeBSD.

В установке *Nginx* возможны два варианта развития событий: сборка из исходных текстов и установка из пакета. Первое происходит стандартным образом: *./configure*, *make* и *make install*. Команда *./configure* в данном случае принимает множество различных параметров – вот самые важные из них:

» *--with-http_ssl_module* позволяет реализовать работу HTTP-сервера по протоколу HTTPS.

» *--without-http_proxy_module* запрещает сборку прокси-модуля HTTP-сервера.

» *--without-http_gzip_module* отменяет сборку модуля компрессии ответов HTTP-сервера.

Полный список параметров можно найти на официальном сайте проекта или в выводе *./configure --help*.

Собрав *Nginx*, не спешите тут же его запускать: он еще не настроен и не знает, что нужно делать для решения нужных вам задач. Об этом мы сейчас и поговорим.

На старт, внимание, марш!

Главный инструмент и помощник в настройке *Nginx* – конечно же, его конфигурационный файл, имя которого **nginx.conf**.

Из чего он состоит? Из секций и директив (впрочем, деление условно), которые могут находиться как внутри секций, так и вне них (отметим, что расположение директивы внутри какой-либо секции еще не означает, что она обязательно не глобальная). По умолчанию (в стандартном файле конфигурации) часть директив закомментирована. В зависимости от стоящих перед вами задач, можно либо раскомментировать их, либо оставить все как есть.

Чтобы правильно настроить *Nginx*, необходимо ответить на серьезный вопрос: для чего мы будем его использовать? Примерное определение можно найти, вспомнив материал первой статьи: *Nginx* хорош в качестве HTTP-сервера и прокси (как web, так и почтового). Вот и все. Но и HTTP-серверы бывают разные – с поддержкой PHP и без нее, поддерживающие SSL и нет, и т.д., и это следует учесть.

Прежде чем переходить к рассмотрению процессов настройки каждого из выделенных нами направлений, скажем несколько слов о глобальных директивах конфигурационного файла *Nginx*.

Глобальными эти директивы называются совсем не случайно. Они относится к работе всего *Nginx* в целом, а не к каким-то отдельным модулям. Так, директива worker_processes указывает максимальное количество рабочих процессов, user – группу и пользователя, от имени которого будут запускаться рабочие процессы, а include позволяет подключать другие файлы. На сегодняшний день директив такого типа у *Nginx* не так уж много – всего одиннадцать.

Секции, которые могут находиться в файле конфигурации *Nginx*, относятся к модулям программы: ngx_http_core_module, ngx_http_access_module или любому другому.

Nginx как HTTP-сервер

Допустим, вы хотите собрать незамысловатый web-сервер, использующийся для «отдачи» пользователям различного статического контента. Что необходимо сделать? В простейшем случае надо:

1 Указать пользователя, от имени которого будет работать наш web-сервер, в глобальной директиве user; количество рабочих процессов (директива worker_connections) и файл, куда будут записываться сооб-

щения об ошибках (директива error_log). Также нужно указать тип передаваемых данных по умолчанию (директива default_type).

2 Подумать над форматом записей log-файлов. Например, можно использовать такой формат записи, где будут отражены адрес пользователя, его браузер и т.п. Это делается с помощью директивы log_format.

3 Помимо указания дополнительных параметров, относящихся к передаче данных (например, tcp_nodelay), нужно с помощью директивы listen указать порты, которые будет прослушивать наш web-сервер (т.е. ждать на них запросов от пользователей).

4 С помощью секции location (модуль ngx_http_core_module) указать выбор нужных конфигураций в зависимости от URI запроса.

В общем виде, файл конфигурации HTTP-сервера, «раздающего» статический контент, может выглядеть так, как показано ниже (указаны только основные директивы).

» Подключаем входящий в дистрибутив *Nginx* файл соответствий расширений файлов MIME-типам:

```
http {
 include conf/mime.types;
```

» MIME-тип по умолчанию, например, text/plain. В нашем случае можно указать application/octet-stream. Стоит также обратить внимание на директивы модуля ngx_http_core_module, чьи имена содержат в себе «client»: client_body_buffer_size, client_body_timeout, client_max_body_size и прочие – их назначение можно уточнить в документации:

```
default_type тип;
```

» Включение (on) следующей директивы позволяет осуществлять сжатие ответов сервера и «тянет» за собой нескольких сопутствующих директив: gzip_comp_level (уровень компрессии), gzip_buffers (буферы, используемые для сжатия). Советую включить, если ваш статический контент не является сжатым сам по себе:

```
gzip on/off;
```

» Разрешим *Nginx* использовать системный вызов sendfile(). Рекомендуется к использованию:

```
sendfile on/off;
```

» Включение или отключение алгоритма Нэгла (Nagle). См. man 7 tcp:

```
tcp_nodelay on/off;
```

» Время, по истечении которого keepalive-соединение будет разорвано сервером:

```
keepalive_timeout число;
```

Следующее, что мы должны сделать – открыть секцию server.

» Укажем порт, прослушивающий *Nginx*, обычно 80:

```
server {
 listen номер порта;
```

» Директива, определяющая имя виртуального сервера:

```
server_name имя;
```

» Если следующая директива находится в положении on, то в поле Content-Type заголовка ответа будет добавлена информация о кодировке, см. ngx_http_charset_module:

```
charset on/off;
```

» Данная секция (хотя корректнее ее называть все же директивой) определяет конфигурацию для всех запросов, удовлетворяющих указанному шаблону. Например, можно «приказать» *Nginx* обрабатывать те или иные запросы самостоятельно или, наоборот, передавать их back-end'у. Секции location – в сущности, главное в настройке *Nginx* как HTTP-сервера. Понятно, что нужно создать отдельную секцию (простите, директиву) location для обработки каждого типа запроса:

```
}
location шаблон URI запроса;
```

Следуя указанным выше принципам, мы получим простейший, но зато корректно функционирующий web-сервер.

Настройка Nginx как прокси-сервера

Здесь многое идентично предыдущему случаю, но есть «усложнение»: директивы proxy_pass, proxy_connect_timeout, proxy_buffer_size и т.п. (модуль ngx_http_proxy_module). Как вы наверняка уже поняли, эти директивы нужно указывать в секции(-ях) location. Давайте попробуем собрать почтовый прокси-сервер: это самый интересный вариант. Чтобы все получилось, необходимо передать сценарию configure ключ --with-mail на этапе настройки исходных текстов. Принцип действия здесь прост: *Nginx* начинает работать с пользователем, «уточняет» корректность идентификационных данных у сервера авторизации, а затем приступает к передаче данных в оба «конца»: от пользователя к POP3/IMAP-серверу, координаты которого он получил у сервера авторизации, и обратно. Таким образом, нам потребуется указать порты, которые будет «слушать» *Nginx*, и задать адрес сервера авторизации (директива auth_http).

Каким образом нужно отредактировать конфигурационный файл *Nginx*, чтобы он мог работать почтовым прокси-сервером? Давайте поступим вот как.

» Укажем URL HTTP-сервера авторизации:

```
# нас интересуют модули ngx_mail_core_module,
# ngx_mail_auth_http_module,
# ngx_mail_proxy_module, ngx_mail_ssl_module.

mail {
 server_name имя;
 auth_http URL-адрес;
```

» Далее можно привести список IMAP-расширений, отдаваемый клиенту в ответ на команду CAPABILITY:

```
imap_capabilities список;
```

Следующие директивы позволяют указать список POP3-расширений и задать расширенные методы аутентификации POP3-клиентов:

```
pop3_auth список;
pop3_capabilities список;
```

То же самое можно сделать и для SMTP:

```
smtp_auth список;
smtp_capabilities список;
```

Далее следует создать необходимое количество секций server, для каждой из которых указать директивы listen, protocol и т.д.

Говоря о настройке *Nginx*, нельзя не упомянуть о виртуальных серверах. Они реализуются просто и изящно – путем создания требуемого количества секций server, в каждой из которых указываются «прослушиваемые» порты и имя виртуального сервера server_name (модуль ngx_http_core_module).

Настроили? Можно запускать. Просто стартуем исполняемый файл *Nginx* от имени суперпользователя. Затем можно посмотреть список запущенных в системе процессов, и если все в порядке, увидеть процессы *Nginx*: главный (master) и несколько рабочих (worker). Как основный, так и рабочие процессы поддерживают управление сигналами (например, SIGQUIT, SIGHUP, SIGWINCH и т.д.). С их помощью можно «заставить» *Nginx* заново обработать файл конфигурации, произвести ротацию журналов, обновить сервер и многое другое.

Заключение

Некоторые читатели возмутятся: где же полные примеры конфигурационных файлов? Их нет; и это было сделано сознательно, т.к. идти по привычному пути не хотелось. А хотелось показать некий концептуальный подход к *Nginx*, не слишком «завязанный» на конкретные директивы и технические нюансы, и продемонстрировать, как профессионально и красиво построен *Nginx* с архитектурной точки зрения; насколько легко его настраивать и работать с ним. Можно отметить, что все директивы конфигурационного файла прекрасно рассортированы по модулям; и те, и другие обладают удобными, интуитивно понятными названиями.

Безусловно, *Nginx* еще есть куда развиваться. Его создатель наверняка добавит в продукт немало примечательных функций; среди прочих экспериментальных возможностей следует упомянуть модуль ngx_http_perl_module, реализующий встроенный интерпретатор Perl. Пожелаем проекту дальнейших успехов, а тем временем будем ждать первую версию продукта, и нормальную документацию к нему. LXF

Попав под колеса индустрии



Джереми Эллисон

Джереми – ведущий разработчик команды Samba, сейчас работает на Google в Сан-Хосе, Калифорния.

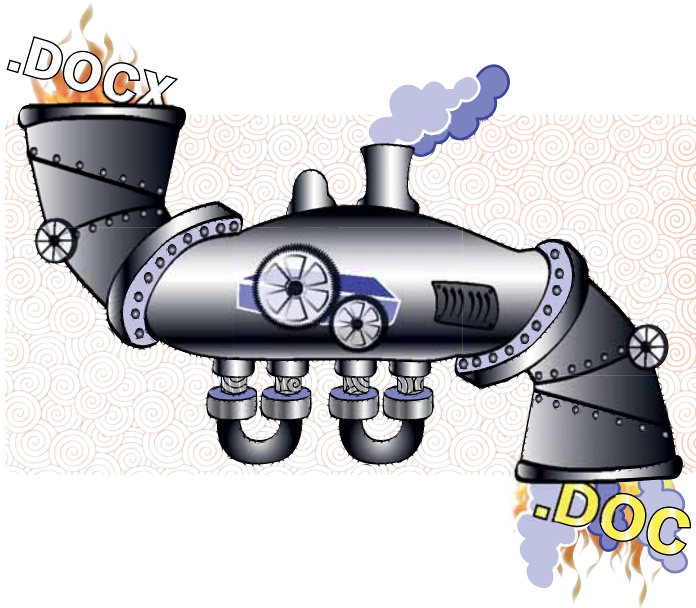
Будучи компьютерным оракулом семьи, легко угодить в переделку. Как-то раз я сидел с друзьями своей жены, которые весело щебетали по-китайски. А я на этом языке умею только считать до десяти, и глубокомысленные беседы мне поддерживать сложно, коль скоро речь не идет об арифметике. Я заскучал, а это всегда штука опасная. «У моей дочки новый компьютер – вы не могли бы помочь настроить его?» – спросил один из друзей. «С удовольствием», – ответил я.

Это был новый ноутбук с установленной Windows Vista. «Вы не могли бы установить какие-нибудь полезные программы, ну, чтобы она могла делать уроки с их помощью?» – попросил этот отец. Предустановленным офисным пакетом была пробная версия *Microsoft Office 2007*, поэтому я скачал *OpenOffice.org* и множество других, более полезных, интересных и находящихся в свободном доступе образовательных программ из сети. Не все они были свободными – я догматик, лишь когда речь идет о свободе ПО на моем личном компьютере, но при настройке программ для ребенка не до проповедей. Я настроил *OOo* на сохранение файлов по умолчанию в форматах *Microsoft Office*, чтобы девочка не страдала насчет обмена файлами с друзьями и учителями.

Весьма довольный собой, я принялся показывать все то новое, что установил на компьютер, «…и у тебя даже есть теперь свободный офисный пакет, чтобы ты могла делать уроки, он совместим с *Microsoft Office*», – гордо возвестил я в конце демонстрации. «Здорово, а можно я посмотрю ту домашнюю работу, которую я уже сделала?» – спросила она. Я направил ее в папку **Documents** (Документы) и, к своему смятению, обнаружил там несколько файлов с расширением не .doc, на которое я рассчитывал, а .docx. Это новый формат документов, OfficeOpenXML, введенный *Microsoft* для *Office 2007*. Поскольку я не пользуюсь *Microsoft Office* (дожидаясь порта под Linux), я совершенно забыл об этом. Пытаюсь открыть эти файлы в *OOo*. Безуспешно. Будучи по натуре человеком аккуратным, я удалил пробную версию *Microsoft Office 2007*, чтобы освободить место на диске. Закачка размером 800 МБ… масса занудных регистраций… и пробная версия установлена заново.

«Я конвертирую все назад, в формат DOC, который использовали предыдущие версии, – сказал я девочке. – Скорее всего, именно эти версии установлены у тебя в школе». Но, прочитав файлы в *Word 2007* и найдя пункт Save As (Сохранить Как) в новой системе меню, я в ужасе убедился, что этот пункт неактивен. «Данная функция доступна только в полной версии *Office 2007*», сообщила подсказка. «Нажмите здесь, чтобы приобрести полную версию». Я решил прибегнуть к более отчаянной мере. Я выделил весь файл и занялся поиском функций Copy (Копировать) и Paste (Вставить). Форматирования можно было лишиться, но у меня хоть сохранился бы полный текст ее сочинения. Copy (Копировать) и Paste (Вставить) тоже оказались неактивны, и появилось то же знаменательное сообщение, что и в прошлый раз.

Заморозить Copy и Paste! Подумайте, это с какого же перепугу и с какой паранойи можно порешить такое на собрании по поводу разработки пробной версии? «Пусть сохраняют файлы в новых форматах. Новые форматы лучше». До такой степени, что выбор пользователя по умолчанию дезактивируется и становится дополнительной функцией, дозволенной только купившим полную версию. Вдобавок новый формат *Office DOCX* еще и несовместим с *OOo*. Это означает, что все документы, созданные за 30 дней пользования пробной версией *Office 2007*, будут по умолчанию нераспознаваемы никакими другими программами, пока вы не оплатите доступ к своим же документам, приобретя полную версию пакета.

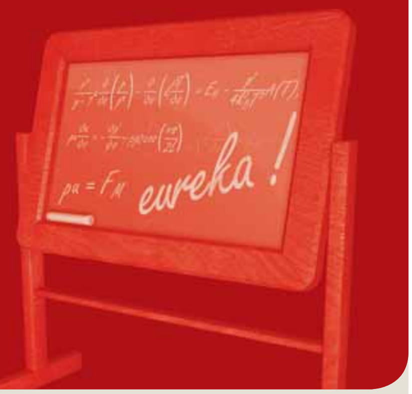


У этой истории счастливый конец: я все же сумел извлечь данные – но стыдобы натерпелся. Девочке пришлось отправить мне документы электронной почтой, и я использовал переводчик *OfficeOpenXML* (формат DOCX), созданный моим другом Майклом Миксом [Michael Meeks] в Novell. Я сохранил старую добрую виртуальную машину SUSE Linux, чтобы исправлять ошибки в Samba на моей прежней работе, и ее хватило для установки переводчика в новую версию *OOo* от Novell и восстановления текста. Кроме того, я получил хороший урок: не стоит делать предположения о том, что нужно людям, или чего они ожидают – в следующий раз буду осторожнее.

Вот вам поучительный пример пользы общественных открытых стандартов. В сотрудничестве с другими, *OpenOffice.org* создал и стандартизировал Open Document Format (ODF). Будь в обоих текстовых редакторах предусмотрена поддержка ODF, было бы совершенно неважно, использует ли дочка моего друга *Microsoft Office*, OOo или еще какой-нибудь текстовый редактор. Возможно, сложное форматирование или презентация и не поддались бы переводу, но ведь большинство людей обычно просто набирают текст, и хотят сделать его попривлекательнее, чтобы начальник их зауважал. Или чтобы получить пятерку за домашнюю работу. Open Document Format – это настолько явное преимущество для всех, что меня удивляет, почему он до сих пор не внедрен повсеместно.

Пожалуйста, помогите формату ODF – поддержите его в своих местных или общенациональных органах власти, в своем сообществе или школе: его нужно популяризировать и поддерживать, иначе не избежать ситуации, когда те, кому не по карману проприетарное ПО, не смогут общаться со своими органами власти или с большинством организаций.

Раз уж я компьютерщик, на прощание подсуну вам задачку. Я придумал еще один способ вытащить девочкины файлы с домашним заданием, без переводчика DOCX. Угадали, о чем я? Как принято в мире истинно открытого кода, давайте вместе придумаем как можно больше способов решения! LXF



Учебники

Наши эксперты помогут вам с любым приложением Linux



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

GPL3: будущее уже с нами

Не страшно, если ученый бьется над неразрешимой задачей.

Пока он осознает, что она не разрешима — он откроет множество интересных вещей.

Ответ на вопрос: «Почему нужно поддерживать фундаментальные науки?»

29 июня 2007 года Фонд свободного ПО (Free Software Foundation или FSF) представил окончательную редакцию третьей версии лицензии GPL (GNU General Public License). Цель *GPL3*, как обычно, абстрактна: свобода для программного обеспечения, но при достижения абстрактных целей приходится решать вполне прикладные задачи, то есть двигаться вперед. На текущий момент цель – патенты.

Пока отношение к *GPL3* у многих настороженное, причем причины этого достаточно очевидны: почти никто не читает тексты лицензий – в лучшем случае, их пролистывают. При этом о качестве лицензии, как и о многих других обыденных вещах, судят по ее распространению. Надо просто подождать, все уляжется само и начнется планомерный переход на *GPL3*, но это будущее.

А сейчас можно попробовать сделать надо собой усилие и прочитать лицензию *GPL3*. Хотя бы один раз, чтобы можно было успокаивать окружающих, говоря: «Я читал и не нашел там ничего дурного – можно использовать». Кстати, мне тоже следует это сделать. Хоть и я доверяю FSF, но простые вещи (лицензия – это даже не бином Ньютона) можно и нужно перепроверять самостоятельно. Чем больше в абстрактную идею вложено мозгов – тем больше от нее практической пользы.

E.M.Baldin@inp.nsk.su

Сегодня в меню...

Совет месяца: В помощь сборщику

Одна из вещей, к которой рано или поздно приходят все пользователи Linux – это самостоятельная сборка ПО. Новичков, привыкших, что установка программы состоит в скачивании файла **setup.exe**, двойном щелчке по иконке и запуске приложения, это слегка шокирует, но компиляция программ из исходных текстов – необходимое зло, так как зачастую другого способа получить новейшую версию просто нет.

Проблема в том, что нормальному пользователю предоставляют не так уж много вспомогательной информации. Если повезет, сработает стандартная троица *./configure*, *make* и *make install* и никакой дальнейшей магии не потребуется. В большинстве случаев этого не происходит – обычно из-за отсутствия зависимостей, т.е. пакетов, необходимых для запуска интересующего вас приложения.

Команда *configure* просматривает вашу систему в поисках таких пакетов и сообщает, что еще необходимо добавить для гладкой установки. Обычно сообщение выглядит примерно так 'FFmpeg AVCodec not found'. В этом случае вы должны запустить менеджер пакетов вашего дистрибутива и поискать недостающий компонент. Понадобится установить как стандартную версию, так и версию для разработчиков, в нашем случае это *ffmpeg* и *ffmpeg-devel*. Если вы не найдете ничего подходящего, придется скачать исходные тексты отсутствующего инструмента из Сети. Даже если *configure* ни на что не пожалуется, вы можете столкнуться с проблемами на стадии *make* – тоже из-за зависимостей.

Чтобы разобраться в происходящем, просмотрите вывод *make* и найдите первое сообщение об ошибке. Если в нем будут слова 'undefined reference', скорее всего, отсутствует некая библиотека. Ее поиск может потребовать некоторых усилий – используйте Google, чтобы найти людей со сходными проблемами или проглядите заголовочные файлы и выясните требования программы.

F-Spot: Хранить

Уезжаете на лето в отпуск? Почему бы не отослать открытку XXI века – пусть завидуют! **Энди Чэннел** покажет, как.





Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется еще со времен Dragon 32

В старые добрые времена мы обычно затаскивали к себе друзей, угощали выпивкой и тут же выуживали пару толстых пыльных фотоальбомов, чтобы освежить веселые воспоминания об еще более старых временах. Теперь все изменилось, и чтобы вспомнить вместе о былых шалостях, можно уже не разоряться на спиртное. Не обязательно даже жить в одной стране с вашими жертвами… ну, друзьями, чтобы всучить им свои фотошедевры. Гип-гип, ура!

На этом уроке мы исследуем одно из лучших приложений для управления фотографиями в Linux – *F-Spot* – и рассмотрим способы его использования для отправки фотографий в большой мир, чтобы ваши близкие могли их видеть. Мы также освоим процесс настройки RSS-ленты фотоальбома, чтобы друзья и родственники были в курсе любых сделанных вами обновлений. Это удобно практически всегда, но во время летнего отпуска особенно: ведь благодаря такой системе вы можете хоть кого заставить позавидовать вашему чудесному отдыху где-нибудь под пальмами (предположим, что выход в сеть есть везде). Считайте это версией почтовых открыток XXI века.

Я выбрал *F-Spot* по нескольким причинам. Во-первых, это само по себе прекрасное приложение. Во-вторых, оно прекрасно сочетается с онлайн-службами публикации фотографий. И в-третьих, оно есть едва ли не во всех дистрибутивах Linux. Если оно не установлено по умолчанию, ваш менеджер пакетов это быстро исправит.

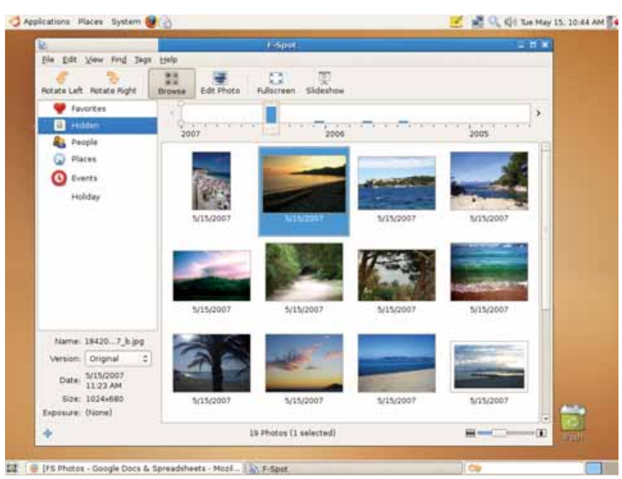
Последняя версия – 0.3.5, и именно она поставляется с последним релизом Ubuntu. Если вам необходимо обновление, посетите сайт проекта (http://f-spot.org) и поищите на странице Downloads [Загрузки] ссылку на ваш дистрибутив. *F-Spot* создан в Mono, и если приложения Mono у вас еще не установлены, вы можете отметить, что этот небольшой пакет в 2,5 МБ требует увесистой загрузки и долгой установки. Но не расстраивайтесь: подождать стоит!

Обеспечив локальные тылы *F-Spot*, разберемся со вторым компонентом данного урока, то есть онлайн-частью – ясно, что мы намерены разместить наши фото в сети, чтобы их получила наша аудитория. Здесь есть три варианта: Photobucket (http://photobucket.com), Flickr (www.flickr.com) и Picasa Web Albums (http://picasaweb.google.com). Все они имеют и плюсы, и минусы. Photobucket популярен, но не полностью интегрируется в *F-Spot*, хотя можно настроить *F-Spot* на отправку фотографий по почте на ваше имя в Photobucket. Flickr хорошо интегрируется, популярен и имеет хорошие настройки безопасности, но пробьет брешь в бюджете, если загружать множество фотографий каждый месяц. Picasa также интегрируется в *F-Spot* и, в отличие от Flickr, предоставляет большой объем бесплатного пространства для размещения, но не богата настройками безопасности.

Работа с двумя последними службами примерно одинакова, и я покажу, как это делается. Первый шаг – регистрация в одной из служб (или в обеих!). Каждая имеет три опции, с неизбежными ограничениями. Ограничение во Flickr – допускается загружать не более 100 МБ фотографий в месяц, а максимальный размер фотографии – 5 МБ. Вроде не проблема, но если у вас особо современная камера, то на 2 МБ потянет каждое фото, и если вы планируете загрузить за выходные более 50 снимков, то, вероятно, лучше взять учетную запись Pro за $25 и отменить все ограничения на скорость и объем загрузки. В Picasa нет ограничений на месячную загрузку, но вся библиотека должна влезать в 1 ГБ. Увеличение этого объема до 6,5 ГБ стоит $25 в год, если уж ваш фотоархив столь огромен, но для наших учебных целей вполне достаточно бесплатной учетной записи.

Изображения на вход...

Первым делом надо занести несколько изображений в *F-Spot*. В отпуске вы будете ежедневно добавлять фото с цифровой камеры, но для



› *F-Spot* – знаток управления большими и малыми коллекциями

» **Месяц назад** Совместный доступ, но не к фото, а к файлам, через FTP или Интернет.

фото в сети

начала добавим их из других источников. Смысл в том, чтобы настроить приложение до того, как упаковать ноутбук и отправиться в дорогу. Вам не захочется заниматься ерундой с приложением и сетевыми службами (особенно если вы платите за интернет-подключение в отеле) в пути. Так что припасем несколько изображений где-нибудь в вашем компьютере.

Вы можете заметить, что после установки *F-Spot* создает каталог с именем **Photos** в вашем домашнем каталоге, [В Mandriva 2007.1 после установки программа вообще ничего не создала и постоянно жаловалась, что не может найти целевой каталог. После его создания вручную все заработало, – прим.пер.] в котором он предпочитает хранить вашу коллекцию. Можно или бросить группу фотографий прямо сюда, или позволить приложению переместить их. Но сначала надо запустить *F-Spot*. Обычно это делается щелчком на иконке, спрятанной в пункте Графика главного меню рабочего стола.

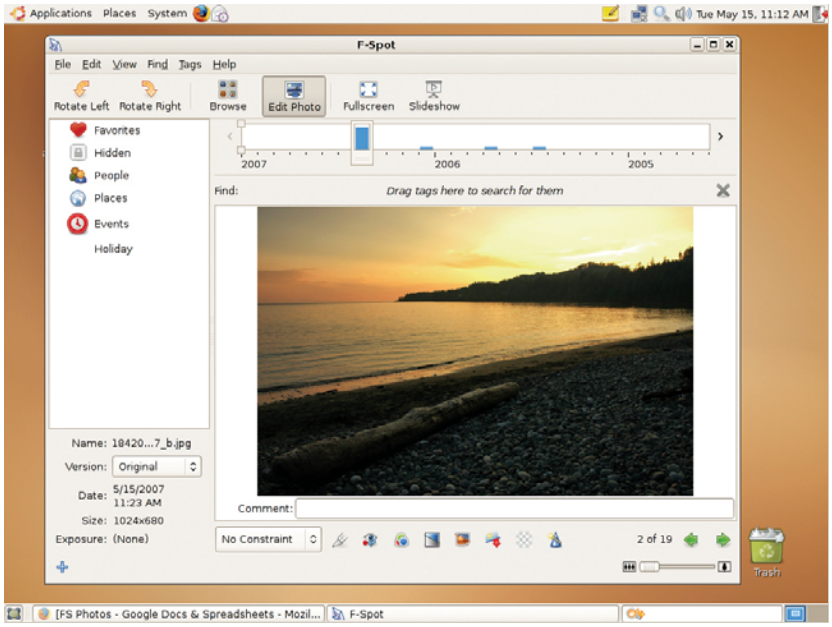
Перед вами откроется простое окно (например, как внизу слева), четко разделенное на три области. В верхней части окна расположены строка меню и панель инструментов. Последняя содержит (слева направо) средства для вращения изображений, кнопки Browse [Просмотр] и Edit [Изменить] и кнопки полноэкранного режима слайдшоу. Внизу слева расположено окно меток (тегов) и место для отображения информации о картинке. Самая большая часть интерфейса – окно Просмотра, в нем видны миниатюры отобранных изображений из каталога **/Photos**. Вы можете изменить размер этих миниатюр, используя небольшой ползунок в правом нижнем углу окна.

Над главным окном просмотра находится интересный виджет-календарь, который становится все полезнее по мере разрастания коллекции. В принципе это гистограмма, и чем больше столбец, тем больше фотографий в этом месяце – легко определить, когда вы были наиболее продуктивны: вероятно, пики этой гистограммы придутся на рождество, день рождения и время отпуска. Щелкните на столбике для быстрого перехода к этому месяцу, и его изображения отобразятся в окне просмотра.

При первом запуске приложения вы увидите, что на панели инструментов выбрана кнопка Просмотр, но поскольку вы еще ничего не импортировали, то главная панель пуста. Исправим это: нажмите Файл > Добавить и перейдите к месту расположения ваших первых изображений. По умолчанию *F-Spot* копирует все выбранные изображения в каталог **/Photos**, не удаляя оригиналы; об этом важно помнить, перейдя к импорту изображений с камеры, потому что вам придется вручную очищать карточку памяти после копирования ее содержимого. Чтобы изменить это, уберите флажок Копировать файлы в папку Photos. Вам, конечно, следует также убрать его, если вы уже скопировали изображения в папку **Photos** и теперь импортируете их.

Для импорта, нажмите выпадающий список Откуда добавить и Выберите папку для запуска стандартного диалога выбора файлов. Теперь вы можете перейти к своим изображениям, нажать кнопку OK, затем щелкнуть на кнопке Импорт и подождать, пока не появятся изображения. В зависимости от размера вашей библиотеки, это может занять от нескольких секунд до минут. Импорт с камеры – точно такой же процесс, только в выпадающем списке вы выбираете свою камеру (Linux распознает ее сразу после подключения) вместо каталога.

Одна из интересных функций F-Spot – присвоение изображениям меток во время импорта при помощи выпадающего списка Добавить метку. Имеется несколько предопределенных меток, и можно также создать свои собственные, для вашей семьи, деятельности или



› *F-Spot* содержит небольшой набор инструментов для редактирования изображений.

предпочтений. Метки позволят вам позднее сортировать изображения по каким-то критериям, так что они очень полезны. Более того, можно преобразовать метки *F-Spot* в метки Flickr при экспорте, и это здорово.

Главная панель инструментов интуитивно понятна. Например, вы выделяете изображение в главном окне, чтобы увидеть информацию о нем, включая имя, дату и время создания, а также параметры экспозиции. Выбрав изображение, нажмите кнопку Повернуть влево или Повернуть вправо, чтобы поменять ориентацию с горизонтальной на вертикальную (или наоборот), или сделайте более серьезные изменения, нажав кнопку Редактировать снимок.

F-Spot не намерен бросать вызов *GIMP* по части редактирования изображений, но содержит хороший набор инструментов, необходимых для их подготовки к показу в Интернете или отправки на печать. У левого края панели инструментов Редактирование (которая появляется у нижнего края обрабатываемого изображения) находится выпадающий список формата – перечень стандартных размеров вроде 4x6 и 8x10 (дюймов – *F-Spot*, как и многие области традиционной фотографии, все еще не перешел на метрические измерения). Он пригодится, если ваш фотопринтер рассчитан на стандартные размеры. Если вы выберете один из них, а затем станете выделять изображение, будет подсвечиваться область с выбранным соотношением сторон. Определившись с выбранной областью, щелкните на кнопке Кадрировать по выделению, она находится рядом с выпадающим списком. Если вы кадрируете чисто по эстетическим причинам, а конкретный размер вас не заботит, просто убедитесь, что в списке выбрано Свободное выделение и затем вновь выберите желаемую часть изображения и нажмите кнопку Кадрировать по выделению.

После кадрирования изображения, выпадающий список Версия в информационной панели изменится с Исходная на Версия (1). Это означает, что на любом этапе редактирования вы можете вернуться к оригинальному изображению или сравнить его с полученным. Учтите, сохраняется не каждая стадия редактирования, так что если вы кадрировали изображение, затем изменили цветовой баланс, яркость »

Скорая помощь

Незачем загружать в сеть большие фото. Измените их размер до 800x600, то-то друзья порадуются.

» или другие элементы во второй иконе слева, пошаговый откат станет невозможным: все или ничего. Решением является начинать с создания новой версии (Файл > Создать новую версию...), а затем выполнять главные преобразования над ней. Число создаваемых версий не ограничено, и вы можете создать множество версий одного изображения, не утратив оригинал.

Другие кнопки редактирования в этом экране – преобразование в черно-белый режим или сепию, подстройка вращения, эффект мягкого фокуса (осторожно! он не так уж мягкой) и инструмент автокоррекции, пытающийся создать сбалансированное изображение, играя с настройками цвета. У правого края панели инструментов находится виджет-лупа, увеличивающий или уменьшающий основное изображение, и кнопки навигации Предыдущий/Следующий снимок.

Наконец, в главной панели инструментов есть опции Во весь экран и Слайд-шоу. Первая убирает интерфейс пользователя *F-Spot* так, чтобы вы могли не отвлекаясь смотреть на изображение. Если в полноэкранном режиме подвигать мышью, то в нижней части окна появится небольшая панель инструментов, позволяющая вращать изображение, запустить слайд-шоу, выбрать переход между слайдами и передвигаться по вашей библиотеке, используя кнопки Назад и Вперед. Кнопка Слайд-шоу отобразит фотографии вашей библиотеки во весь экран или в родном разрешении (если изображение имеет меньшее разрешение, чем экран).

Теперь наши изображения занесены в приложение, мы выполнили подстройку, сделав место нашего отпуска теплее (совет: нажмите кнопку Скорректировать цвета снимка и немного увеличьте параметр Температура), и пора отправить их в сеть для тех наших друзей, кто не в отпуске.

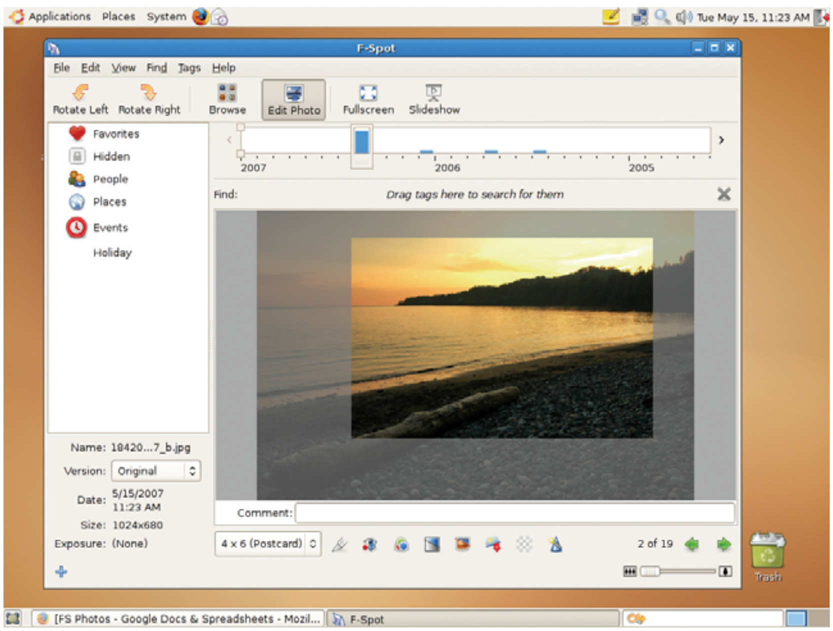
Скорая помощь

Используйте сайты для размещения фотографий как открыток, вместо почтовых, но не вывешивайте чересчур личные фото!

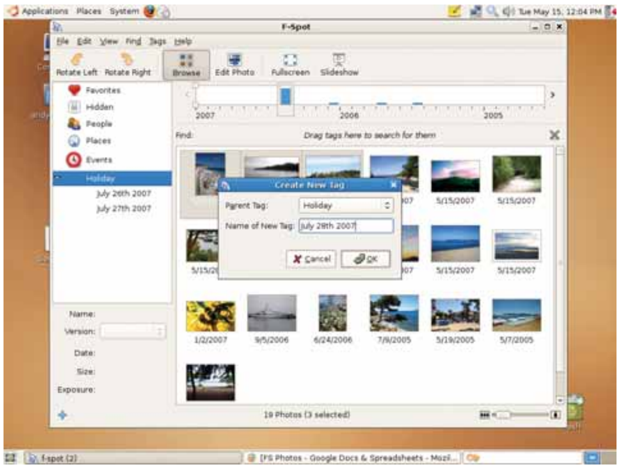
Изображения на выход...

Экспорт изображений из *F-Spot* не сложнее импорта. Имеется на выбор: экспорт в каталог, создание web-галереи или использование одной из четырех web-служб. Как указывалось выше, мы будет выводить прямо во Flickr и Picasa, так что заведите себе учетную запись перед тем, как мы двинемся дальше.

Перед экспортом необходимо выбрать изображения, которыми вы хотите поделиться. Как всегда, имеется несколько способов сделать это. Можно, например, создать метку для каждого дня вашего отпуска – выполните Метки > Создать новую метку... и дайте ей имя и/или «Родительскую метку» – а затем дважды щелкните на метке (или щелкните правой кнопкой на метке и выберите Найти), чтобы оставить только соответствующие изображения в окне просмотра.



› **Набор стандартных размеров кадрирования в F-Spot делает привлекательные композиции отчетливыми.**



› **Метки помогут сортировать ваши изображения по любому принципу – здесь я использовал даты; используйте также привычные других пользователям метки.**

Теперь выбирайте из этого подмножества изображения для загрузки на сайт. Отдельное изображение выбирается щелчком на нем; последовательный их набор – щелчком на первом изображении серии, а затем, при нажатой клавише Shift, на последнем; или выберите произвольные несколько изображений, прощелкав по ним при нажатой клавише Ctrl.

Собрав изображения, выберите Файл > Экспортировать > Во Flickr или В Picasaweb.

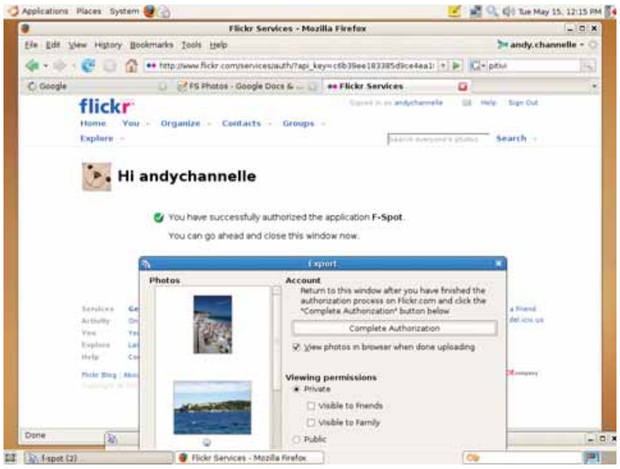
» **Flickr** Когда вы экспортируете что-то во Flickr впервые, необходимо настроить приложение для входа под вашей учетной записью. Для этого нажмите массивную кнопку Авторизоваться. Откроется web-браузер, в котором вы сможете войти во Flickr. Вы должны увидеть сообщение, что *F-Spot* получил права на загрузку под вашей учетной записью. Теперь вернитесь назад в *F-Spot* и нажмите кнопку Завершить авторизацию. Эта процедура выполняется лишь один раз.

Теперь вы готовы к загрузке на сайт. Обратите внимание на настройки безопасности в окне Экспорт, они повторяют настройки вашей учетной записи. Здесь также есть опции изменения размера изображений перед их загрузкой, что полезно, если у вас медленное соединение, вы почти выбрали лимит или считаете, что изображения будут просматриваться только на экране; удаления метаданных (даты, времени, экспозиции и так далее) и экспорта меток. Если вы выбрали последнюю опцию, то приложение преобразует ваши метки *F-Spot* в метки Flickr. Если такая метка во Flickr не существует, то она будет создана; если она уже есть, то будет просто применена к изображению. По умолчанию, сделанные здесь изменения применяются к каждому изображению в корзине загрузки. Однако вы можете изменить параметры отдельных изображений, выбрав их в окне предварительного просмотра слева.

Когда все будет готово, нажмите OK и подождите, пока изображения будут отосланы на сервер; потом можно приступать к следующей стадии.

» **Picasaweb** Первый экспорт в Picasa Web Albums заставит вас войти в вашу учетную запись. В верхней части окна экспорта нажмите кнопку Добавить рядом с пунктом Галерея: и введите имя пользователя и пароль. Если у вас несколько учетных записей Picasaweb, то вы можете добавить каждую из них, а затем выбрать желаемую, используя выпадающий список Галерея: перед экспортом.

Ниже этого раздела вы увидите информационное пространство, где показано, сколько места используется учетной записью, а еще ниже – выпадающий список доступных альбомов. Альбомы в Picasaweb могут быть общедоступными или нет, так что используйте настройки благоразумно. Вы можете добавить новый альбом (и установить его публичный статус), щелкнув на большой дружественной кнопке Add [Добавить] и введя имя, описание и настройки приватности. Снизу к этому окну прилегают опции изменения размера изображения и его Автовращение, означающее, что приложение пытается сообразить,



› Вам нужно авторизовать *F-Spot* лишь однажды, чтобы разрешить загрузку на сайт.

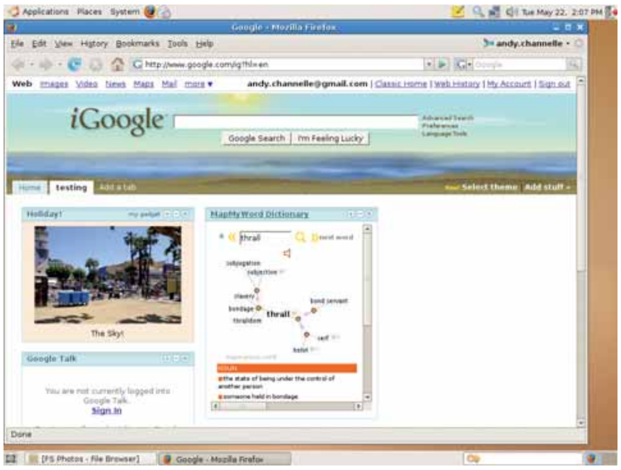
портрет это или ландшафт, а затем соответствующим образом его повернуть. После настройки корзины экспорта нажмите OK и начните загрузку на сайт.

Выставка изображений...

Теперь, когда ваши прекрасные фотографии сохранены в сети Интернет, можете выслать электронным письмом адрес сетевой библиотеки своим друзьям, и они помчатся на Picasa или Flickr (согласно выданным правам) смотреть ваши последние снимки. Но… минуточку! Вдруг ваши друзья ленивы? Не легче ли и эффективнее дать им точно знать, когда вы обновили библиотеку, чтобы им не пришлось проверять библиотеку постоянно?

На наше счастье, какие-то умные головы специально для этого изобрели RSS (аббревиатура расшифровывается как Rich Site Summary [Подробная сводка по сайту] или Really Simple Syndication [Действительно простая синдикация] в зависимости от того, у кого вы спрашиваете). Что важнее, и Picasa, и Flickr имеют средства настройки RSS-ленты, а значит, ваша аудитория может «записаться» в вашу библиотеку и будет автоматически уведомляться о новых поступлениях.

Главное отличие Flickr от Picasa в том, что первый может запретить доступ к вашим фотографиям определенному кругу друзей и знакомых, что определяется вами в приложении. Реальная сила этого заключается в том, что Flickr может работать и с друзьями, и с семьей. Например, серия чисто пейзажных снимков вашего отпуска помещается в общедоступный профиль Flickr «public». Более личные снимки



› При помощи RSS-читалки, семья и друзья могут следить за вашими приключениями на отдыхе почти в реальном времени.

(их-то я и называю почтовыми открытками XXI века) отправляются в раздел «друзья», а узко семейные фото резервируются исключительно для родни. Очевидно.

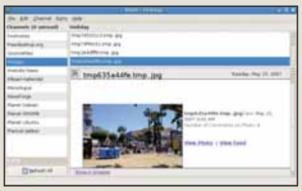
Picasa, наоборот, делает общедоступным все, с оговоркой, что вы сами укажете, сделать ли ваши изображения доступными по поиску. Это означает, что любой посетитель должен сначала угадать или раздобыть таинственный URL вашей библиотеки изображений (к примеру, http://picasaweb.google.co.uk/andy.channelle/Animals?authkey=u-jwWtlu_3Q), чтобы суметь просмотреть ваши снимки. Вы можете, естественно, сделать часть библиотек общедоступными ('public'), тогда ваши изображения будут доступны для поиска и просмотра любому.

Чтобы этим воспользоваться, вы отсылаете web-адрес своего Flickr-потока или web-альбома друзьям и родственникам, а те входят и ищут маленькую кнопку с названием RSS. Они копируют этот адрес в свою RSS-читалку (лично у меня *Blam*), и RSS-читалка периодически проверяет наличие обновлений. Заметив любые изменения, она моментально загрузит новые фото, и ваши люди смогут просмотреть их через несколько минут после того, как вы их разместили. Это быстрее, чем почтовые открытки, дешевле (возможно), чем мультимедийные сообщения с мобильного телефона и забавнее, чем семафор. Только убедитесь, что вы сохранили лучшие снимки для ночного слайд-шоу в Linux, приправленного парой бутылочек красного вина или бокалов сангрии, когда вернетесь обратно. LXF



› Picasa – хорошая альтернатива Flickr, особенно если вы загружаете сотни фото для просмотра только на экране.

Даешь гаджет-фото!



› Родные и друзья могут подписаться на ваш 'photocast'.

Если ваша аудитория – пользователи персональных страниц Google (iGoogle), вы можете создать механизм, способный отображать ваши фотографии на их домашних страницах. Для начала войдите на вашу собственную страницу iGoogle и выберите сверху Add Stuff [Добавить элемент]. Щелкните на My Community [Мое сообщество], а затем Choose to make a new gadget [Выбрать создание нового гаджета]. Нам нужен гаджет Framed Photo [Фото в рамке], поэтому щелкните на этой опции и введите детали до семи фотографий. Эти изображения определяются по их URL, так что вы можете брать их с любого сетевого источника, включая Flickr и Picasa Web Albums.

После выбора изображений добавьте их описание, а затем нажмите Create Gadget [Создать гаджет]. Гаджет добавится к вашей домашней странице, и вы сможете ввести список электронных адресов в окне уведомлений; каждый из этого списка получит возможность добавить его себе. Умно! И вы опять-таки можете выбрать, создать публичный гаджет или личный.

» Через месяц С фотографиями вообще-то все! На очереди видео.

Магия PHP! Приложите руки к созданию web-приложения, использующего базу данных.

Akelos: Долой

Не думайте, что у Ruby есть Rails, а у PHP нет. С помощью Akelos, динамические программы будут лететь у вас из-под пальцев, утверждает **Дэн Фрост**.



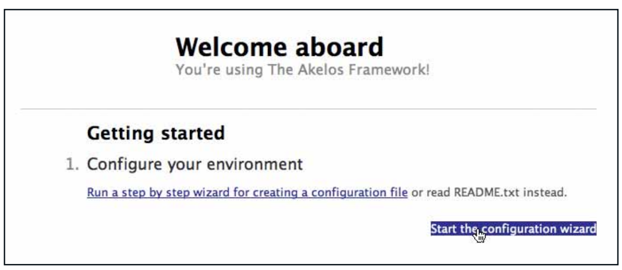


Наш эксперт

Дэн Фрост
Дэн работает в Брайтоне техническим директором 3ev, это агентство по разработке web-приложений. Он работал на проект CMS TYPO3, а в настоящий момент занимается CMS-приложением 'Involve' для 3ev.

Ruby on Rails взял штурмом мир программистов, вызвав у тех из нас, кто все еще пишет на PHP, жуткую зависть к такому элегантному методу работы. Если вы еще не видели пятнадцатиминутный ролик *Look at what I'm not doing* («Смотри, чего я НЕ делаю») на www.rubyonrails.org – посмотрите непременно: он запросто сагитирует вас плюнуть на PHP и перейти на Ruby.

Вся магия Rails – в метапрограммировании. Немалая часть кода создается за вас, так что вам не приходится всякий раз вспоминать, где какой класс находится и куда поместить такую-сякую функцию. Это сильно упрощает работу в рамках архитектуры: обучение ускоряется, и надо меньше задумываться в процессе работы. Таков Rails; но ведь у сообщества PHP есть и собственные системы разработки. У каждой из них свои сильные стороны, и Akelos – один из самых сильных среди них. Можно сказать, что Akelos – это Rails для PHP, насколько такое вообще возможно. Большая часть архитектуры системы сохранилась, за исключением тех моментов, которые на PHP лучше сделать иначе. Akelos написан так, что он совместим и с PHP 4, и с PHP 5, так что его можно установить практически на любом сервере. Это было одним из важных аргументов в пользу Akelos в нашей компании, так как, хотя мы стараемся в основном использовать PHP 5, иногда нам приходится разрабатывать приложения для работы на старом оборудовании или же для клиентов, которые просто не хотят обновлений.



› **Рис. 1. Распакуйте исходный код и откройте браузер – на установку трёх баз данных больших усилий не потребуется**

Все это делает Akelos идеальным для быстрого развертывания приложений, которые должны работать где угодно. На нашем уроке мы рассмотрим его и создадим простенькое web-приложение, чтобы вы поняли, как использовать каркас на полную катушку. В итоге вы получите полноценную рабочую программу, но учтите, что на серьезное рассмотрение у нас здесь места мало – мы только пройдемся по верхам. Советую самостоятельно зарыться в код и разобраться, на что он еще способен!

Чего вы не делаете

Благодаря метапрограммированию, вы можете изучить Akelos очень быстро, потому что он очень многое сделает за вас. При изучении других сред программирования вам нужно было помнить, где расположить классы каких типов, какие функции какие классы реализуют, и как применять объекты каждого из типов. На Akelos вам не нужно изучать, что такое контроллер и куда его засунуть: вы можете предоставить его генерирование программе. Она умеет создавать целые семейства классов и файлов при помощи команды scaffold.

Давайте установим Akelos и попробуем в деле. Вам понадобится сервер Apache с MySQL, SQLite или PostgreSQL. Возьмите Akelos с www.akelos.org или из каталога **Magazine/Akelos** на приложенном диске. Распакуйте файл в каталог (он будет называться **akelos_framework**), расположенный внутри корневого каталога web-сервера. В целях нашего урока назовем этот каталог **akelos**, так что теперь вы можете открыть его в браузере по адресу http://localhost/akelos.

Выполните инструкции, чтобы завершить установку. Вам понадобится настроить три базы данных – одну для разработки, одну для эксплуатации и одну для тестирования. Для тонкой настройки вам может потребоваться изменить первую строчку файла **/scripts/generate** и некоторых сценариев так, чтобы она указывала на исполняемый файл PHP для работы в командной строке:

```
#!/usr/bin/env php
```

В дальнейшем я предполагаю, что вы установили систему в каталог с названием **akelos**, так что URL для доступа к нему должен быть http://localhost/akelos/public. Если у вас возникли сложности с просмотром примеров в браузере – проверьте, как начинается правило RewriteBase в файле **public/.htaccess** – оно должно содержать путь к каталогу **public**.

Итак, мы можем начинать. С помощью вашей любимой оболочки командной строки перейдите в каталог с распакованным приложением и выполните команду generate:

```
cd /path/to/untarred/akelos/
./scripts/generate
```

Вы увидите некоторое описание, что делает generate и как ее использовать. На следующей странице, во врезке «Сценарии» вы найдете список команд, которые она умеет выполнять. Пока запустите

```
./scripts/generate scaffold Post
```

а затем откройте в браузере адрес http://localhost/akelos/public/post (не забудьте изменить адрес, если вы установили Akelos в какую-то другую директорию). Вы увидите экран со ссылкой для создания нового Post'а, довольно пустынный, однако кнопка Create New Post там есть.

Лишний код!

LINUX FORMAT На DVD » Пре-релиз Akelos

Кликните на нее и введите какие-нибудь данные (Рис. 1). После сохранения записи вы сможете увидеть ее на http://localhost/akelos/public/post, а также сможете изменить или удалить ее, пользуясь соответствующими ссылками (Рис. 2).

Команда scaffold генерирует для вас приложение, уже имеющее базовый набор функций, ожидаемый от web-приложения: создание, редактирование и удаление записи в базе данных из браузера. В этом вся прелесть метапрограммирования – большую часть работы делать не надо!

Но что все-таки произошло? А вот что: генератор кода создал для вас каркас приложения, включающий контроллер, помощник и несколько видов. В нашем случае он так же создал простой класс model, состоящий из названия и описания, что и позволило вам заполнять эти поля. Одной командой – generate scaffold – вы выполнили основную часть работы по написанию приложения для создания и редактирования записей базы данных.

Взгляд на вид

Вид – это такая штука, которую видит пользователь. Это буквально и есть внешний вид программы. Все виды вашего приложения должны располагаться в папке **app/views**. Когда вы открываете страницу приложения Akelos, соответствующий вид определяет, как она выглядит. Например, при открытии http://localhost/akelos/public/post/listing вызывается шаблон **views/post/listing.tpl**. Когда вы щелкнете на какой-нибудь элемент списка, в работу вступит **view.tpl**.

Откроем **/views/post/listing.tpl** и рассмотрим его содержимое. Вы увидите, что шаблоны в основном состоят из HTML и PHP, однако в дополнение к ним используется набор специальных тэгов [sintag], позволяющих избежать повторов в коде. Систему тэгов разработала та же команда, которая написала Akelos. Для выделения тэгов они использовали фигурные скобки – { и }.

В файле **listing.tpl** можно найти следующий участок кода:

```
{loop posts}
<tr {?posts_odd_position}class="odd"{end}>
{loop content_columns}
<td class="field"><?php echo $post->get($content_column)?></td>
{end}
...
```

Первая строка начинает цикл по списку объектов Post, а четвертая строчка выводит значение очередного свойства текущего объекта. Попробуйте заменить текст от {loop content_columns} до {end} двумя строчками ниже на

```
<td class="field"><b>{post.name}</b></td>
<td class="field">{post.description}</td>
<td class="field">{post.created_at}</td>
<td class="field">{post.updated_at}</td>
```

Все, что мы изменили – это сделали индивидуальную обработку каждого из полей и добавили жирный шрифт для имени записи, однако это неплохая иллюстрация использования спецтэгов. Чтобы понять, зачем они нужны для некоторых задач в PHP, откройте файл **views/post/compiled/listing.tpl.php**. Посмотрите, на что был заменен один-единственный тэг {look posts}:

```
<?php
empty($posts) ? null : $post_loop_counter = 0;
empty($posts) ? null : $posts_available = count($posts);
if(!empty($posts))
   foreach ($posts as $post_loop_key=>$post){
       $post_loop_counter++;
       $post_is_first = $post_loop_counter === 1;
       $post_is_last = $post_loop_counter === $posts_available;
       $post_odd_position = $post_loop_counter%2;
?>
```

На этом можно пальцы вывихнуть! Если вам придет охота самим организовать цикл на PHP, ничто не мешает это сделать, но система спецтэгов экономит время.

Как насчет дизайна приложения? Главная HTML-страница вашего приложения – **app/views/layout/post.tpl**, а таблица стилей по умолчанию для каркаса Akelos – **public/stylesheets/scaffold.css**. Файл CSS подключается к шаблонам строкой

```
<?php echo $asset_tag_helper->stylesheet_link_tag('scaffold') ?>
```

Можно редактировать файл **scaffold.css** напрямую, но почему бы не разработать для вашей программы свой собственный стиль? Создайте файл **/public/stylesheets/myapp.css** и внесите в него следующие определения:

```
h1 {
border-bottom : thin solid black;
padding : 3px;
}
```

Осталось добавить в заголовок HTML файла **views/layout/post.tpl** следующую строчку:

```
<?php echo $asset_tag_helper->stylesheet_link_tag('myapp') ?>
```

Конечно, вы можете как использовать один и тот же CSS-файл для всех контроллеров приложения, так и определять для каждого из них свой собственный файл и внешний вид.

Контролируем контроллер

Теперь рассмотрим *контроллеры*. Они определяют, что пользователь увидит и когда, а также соответствие URL-адреса и действия, которое можно выполнить по этому адресу. Когда вы посещаете **book/edit/123/**, Akelos определяет, какой контроллер вы хотите использовать, какая операция контроллера была вызвана и к какой записи его нужно применить. В этом примере контроллером будет book, операцией – edit, а номер записи 123.

Откроем **app/controller/post_controller.php** и пробежимся взглядом по его структуре:

```
class PostController extends ApplicationController
{
   // Index is the default action
   function index() { /* нечто... */ }

   // the listing function
   function listing() { /* нечто... */ }
```

»

Скорая помощь

Как вы заметили, иногда я говорю об объектах Post (с заглавной) или о таблицах posts (во множественном числе и нижнем регистре). Akelos пытается подсказывать человеку, используя написание с заглавной для моделей – например, Post – и множественное число – для таблиц.



› Рис. 2. Почти без усилий мы создали средство управления записями.



› Рис. 3. Ваши новые записи – или задачи, или что еще – можно редактировать отсюда.

```
»
    function show() { /* нечто... */ }

    function edit() { /* нечто... */ }

    ...
}
```

Каждая функция контроллера соответствует некому действию, которое может предпринять пользователь – получить список записей, просмотреть их или отредактировать. Сгенерированный нами каркас имеет набор основных операций, свойственных web-приложению, но вы можете добавить к контроллеру все что угодно. Для этого сначала добавляется функция:

```
function hello() {
    // вам совершенно не обязательно что-то тут писать
}
```

После создания функции нужно добавить TPL-файл с тем же именем в каталог **app/views/post/**, так что создадим **app/views/post/hello.tpl** и поместим в него следующую строчку:

```
<h1>Hello, World!</h1>
```

А теперь зайдите на http://localhost/akelos/public/post/hello. Возможно, именно с этого нам бы и следовало начинать, однако с Akelos «Hello World» написать сложнее, чем полноценное web-приложение!

Функции контроллера могут выводить текст напрямую, без использования шаблонов – через функцию render(). Разовьем нашу функцию и обновим страницу в браузере:

```
function hello() {
    $this->render(
        array(
            'text' => 'Hello, World - from inside the controller'
        )
    );
}
```

Этот вызов игнорирует шаблон, выводя данные прямо из контроллера, и его можно использовать для функций Ajax, как мы увидим в дальнейшем.

Наконец, операции контроллера можно перенаправлять на другие операции. Изменим функцию hello() еще раз:

```
function hello() {
    $this->renderAction('listing');
}
```

Теперь, когда вы зайдете на **public/post/hello**, вас перенаправят на страницу со списком записей, точно так же, как это происходит при обработке операции index(). Если вы хотите узнать, на что еще способен контроллер, обратите внимание на функции класса **lib/AkActionController.php**.

Про инсталляторы

Вместе с Akelos вы получаете мощный инструмент управления таблицами баз данных, включая поддержку изменений и обновлений структуры. Это может сэкономить вам часы возни с SQL-кодом и значительно упростить процесс разворачивания и обновления вашей программы. Команда generate scaffold, которую мы использовали раньше, создала в том числе и файл инсталлятора **app/installers/post_installer.php**. Откроем его и посмотрим:

```
function up_1()
{
$this->createTable('posts', "
            id,
            name,
            description,
            created_at,
            updated_at
    ");
}
function down_1()
{
    $this->dropTable('posts');
}
```

Инсталлятор создает таблицу под названием posts с полями id, name, description, created_at и updated_on. Команда generate scaffold тоже создает таблицу автоматически, но вы можете пересоздать ее вручную:

```
./scripts/migrate Post uninstall
```

(эта команда вызывает down_1(), при этом удаляя таблицу), и вернуть ее обратно с помощью

```
./scripts/migrate Post install
```

(вызов up_1() создаст таблицу заново).

Если после этого вы зайдете на http://localhost/akelos/public/post, то увидите, что все записи исчезли, а таблица была пересоздана.

Как вы поняли, функция up_1() используется для установки, а down_1() – для деинсталляции. Точнее будет сказать, что up_1() используется для увеличения номера версии, а down_1() – для уменьшения. Чтобы разобраться в этом, создадим новую функцию инсталлятора с названием up_2():

```
function up_2()
{
    $this->addColumn('posts', 'extracol');
}
```

Запустите команду install снова и посмотрите, как теперь выглядит таблица в базе данных, с помощью клиента MySQL. Видно, что инсталлятор выполнил команду ALTER TABLE posts ADD extracol, добавив в таблицу новый столбец.

Соответствующие функции down_ служат для отката изменений, так что давайте создадим down_2(), действие которой обратно действию up_2():

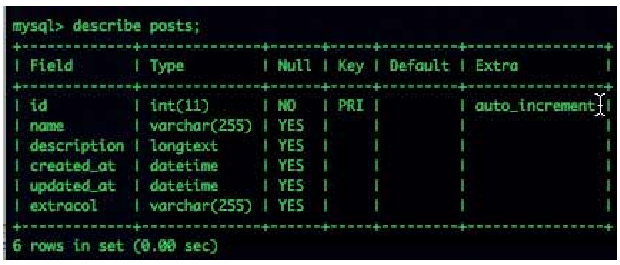
```
function down_2()
{
$this->removeColumn('posts', 'extracol');
}
```

Теперь запустите uninstall, и увидите, что были запущены две SQL-команды:

```
SQL ALTER TABLE posts DROP COLUMN extracol
...
DROP TABLE posts
```

Эти команды соответствуют вызовам down_2() и down_1(). Функции отката должны точно соответствовать функциям установка, чтобы при установке от откате не было никаких ошибок.

Функции createTable(), dropTable и removeColumn() – это просто оболочки для библиотеки доступа к данным AdoDB, которую Akelos исполь-



```
mysql> describe posts;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int(11)      | NO   | PRI |         | auto_increment |
| name        | varchar(255) | YES  |     |         |                |
| description | longtext     | YES  |     |         |                |
| created_at  | datetime     | YES  |     |         |                |
| updated_at  | datetime     | YES  |     |         |                |
| extracol    | varchar(255) | YES  |     |         |                |
+-------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

› **Инсталлятор из состава Akelos упрощает работу в группе: все изменения в базе данных под контролем.**

Скрипты

Вот небольшой список скриптов командной строки из каталога **scripts**.

» **Setup** Используется для установки Akelos из командной строки вместо браузера.

» **Generate** Служит для генерации каркасов, моделей, контроллеров и сервисов.

» **Console** Интерфейс командной строки к окружению Akelos.

» **Migrate** Служит для создания и удаления таблиц баз данных. Он использует файлы из папки **app/installers/**.

Чтобы получить более подробную информацию, запустите команду *generate* без параметров – просто наберите

```
./scripts/generate
```

зует при всех операциях с базой данных. У Akelos есть еще несколько полезных функций, таких как changeColumn(), renameColumn(), addIndex() и removeIndex(). Если вам нужен прямо доступ к объекту AdoDB из инсталлятора, используйте $this->data_dictionary.

Полезно обдумать, какие изменения вносит инсталлятор в процесс разработки и установки приложения. Работающие в группе должны знать, как сложно бывает выкрутиться, когда вносимые в структуру данных изменения конфликтуют между собой – на разрешение таких конфликтов тратится куча времени и нервов. Используя функции повышения и понижения версии базы данных, разработчики могут легко менять структуру и не беспокоиться о конфликтах. Если же конфликт все-таки возникает, достаточно бывает слить две PHP-функции в одну или же написать функцию up_() заново с учетом конфликтных требований.

Маленькие помощники Akelos

А теперь поиграем с *помощниками*, предоставляющими полезные функции, не являющиеся частью модели, вида или контроллера. К ним относятся задачи генерации URL, обработки текста, создания простых меню и построения HTML-тэгов. Вместе с Akelos поставляется целая куча помощников, каждый из которых отвечает за свой тип функциональности. Например, url_helper содержит функции построения URL (например, составление ссылки на какое-то действие определенного контроллера):

```
<?= $url_helper->link_to('Go to the help page', array('action' => 'help') ); ?>
```

Второй ценный помощник – menu_helper. Он умеет генерировать меню возможных операций контроллера, например

```
<?= $menu_helper->menu_for_controllers(array('post'=>array('help', 'list'))) ?>
```

Понравилось? Перечисляя интересных помощников, нельзя не упомянуть про form_helper и date_helper. Очень полезно изучить их возможности (да и остальных тоже), для этого можно заглянуть в **lib/AkActionView/helpers/**.

В добавление к помощникам, содержащим основные функции для использования во всех приложениях, при каждой генерации каркаса создается свой собственный помощник. Например, при создании каркаса Foobar будет сгенерирован foobar_helper, а чтобы вызвать его из кода приложения, можно воспользоваться переменной $foobar_helper. Вернемся к нашему приложению: откройте файл **app/helpers/post_helper.php** и добавьте в него функцию

```
function quick_destroy(& $record)
{
    return $this->_controller->url_helper->button_to('Delete in 1 click',
        array('action' => 'quick_destroy',
        'id' => $record->getId()),
        array( 'confirm' => 'Are you sure?' ));
}
```

А затем в файл **app/views/post/listing.tpl** впишите строчку

```
<td class="operation"><?php echo $post_helper->quick_destroy($post)?></td>
```

Осталось только добавить в контроллер следующую функцию:

```
function quick_destroy()
{
    if(!empty($this->params['id'])){
        $this->post = $this->Post->find($this->params['id']);
        $this->post->destroy();
        $this->redirectTo(array('action' => 'listing'));
    }
}
```

Обновите страницу со списком записей – и увидите новый столбец Delete in 1 click. При щелчке по любой записи в этом столбце, JavaScript выдаст окно с вопросом, уверены ли вы, а после ответа «Да» запись будет удалена без всяких проволочек.

Ajax, но без JavaScript

Технология Ajax позволяет загружать фрагменты HTML на страницу после ее отображения и обрабатывать формы и команды, не перезагружая заново всю страницу целиком. Это стало очень популярным, поскольку так проще и пользователям, и разработчикам. Rails сделал Ajax еще более популярным, так как скрывает большую часть кода на JavaScript. И, как вы можете ожидать, Akelos умеет делать то же самое. Начнем с добавления следующей строки в HTML-заголовок файла **app/views/layouts/post.tpl**:

```
<?= echo $asset_tag_helper->stylesheet_link_tag('prototype') ?>
```

Помощник asset_tag_helper используется для подключения JavaScript, CSS и других внешних файлов, и добавит любой JavaScript из каталога **public/javascripts/** таким же образом.

Затем создадим операцию контроллера ademo (см. раздел «Контролируем приложение» на стр. XX, если вы забыли, как это делается). Кроме того, нам понадобится файл **app/views/post/ademo.tpl**:

```
<h1>Quick AJAX example</h1>
<?= $prototype_helper->link_to_remote('>Load the listing – by AJAX!', array('url' =>
array('action' => 'getme'), 'update' => 'updateme' ) ); ?>

<div id="updateme" style="padding : 10px; border : thin solid black;">...</div>
```

Ну и, наконец, создадим функцию контроллера с названием getme:

```
function getme() {
    $this->render( array( 'text' => 'Whooa! How easy was that?' ) );
}
```

Откройте в браузере **http://localhost/akelos/public/post/ajax**, щелкните на ссылку, и вы увидите маленький текстовый блок, загруженный на страницу без перезагрузки. Таким же способом можно загрузить на страницу все что угодно – другие страницы, формы, картинки – и вам не придется даже задумываться о JavaScript, потому что за него отвечают функции помощника.

Если вы никогда не имели дела с Ajax, система наподобие Akelos – не самый лучший способ изучить его, потому что он скрывает от вас много тонких деталей. Однако вы можете оценить, насколько быстрее делать вещи именно таким способом.

Будущее Akelos

Мы приближаемся к завершению нашего краткого обзора каркаса, который сторицей отплатит за время, затраченное на его изучение. Мы даже не успели рассмотреть такие вещи, как обработка взаимосвязей в базе данных, но вы уже могли понять силу метапрограммирования и чистоту архитектуры – наследие Rails. В моей фирме Akelos используется уже давно, и мы нашли, что он очень удобен для прототипирования, да еще и позволяет в ноль секунд создавать приложения. Благодаря совместимости с PHP4 и PHP5 он просто идеален в том случае, когда заранее неизвестно, в какой среде придется работать программе.

В данный момент существует небольшое, но активно растущее сообщество посвященное Akelos, и Берми Феррер [Bermi Ferrer], ведущий программист, планирует поработать над документацией и перезапустить web-сайт. LXF

А что внутри app?

Распаковав Akelos, посмотрите на список каталогов. Первый из них (по алфавиту) называется **app**. Именно в нем скрипт *generate* создает файлы. Кроме моделей, видов, помощников и контроллеров, про которые вы уже знаете, здесь же находятся следующие вещи:

» **Installers** Как вы уже знаете, они служат для обновления и отката базы данных.
» **Locales** Пригодятся для создания многоязычного приложения.

В главном каталоге Akelos вы ещё найдёте

» **config** Легко догадаться, что тут лежат файлы конфигурации
» **lib** Тут лежит ядро Akelos
» **Public** Это DocumentRoot для Akelos. Укажите его как DocumentRoot вашего сервера при развертывании рабочей системы.
» **Scripts** Тут вы найдёте скрипты *generate*, *setup*, *console* и *migrate*.
» **Vendor** Содержит библиотеки третьих лиц, в частности, AdoDB и Pear. Благодаря им установка Akelos очень проста: все, от чего он зависит, включено в дистрибутив.

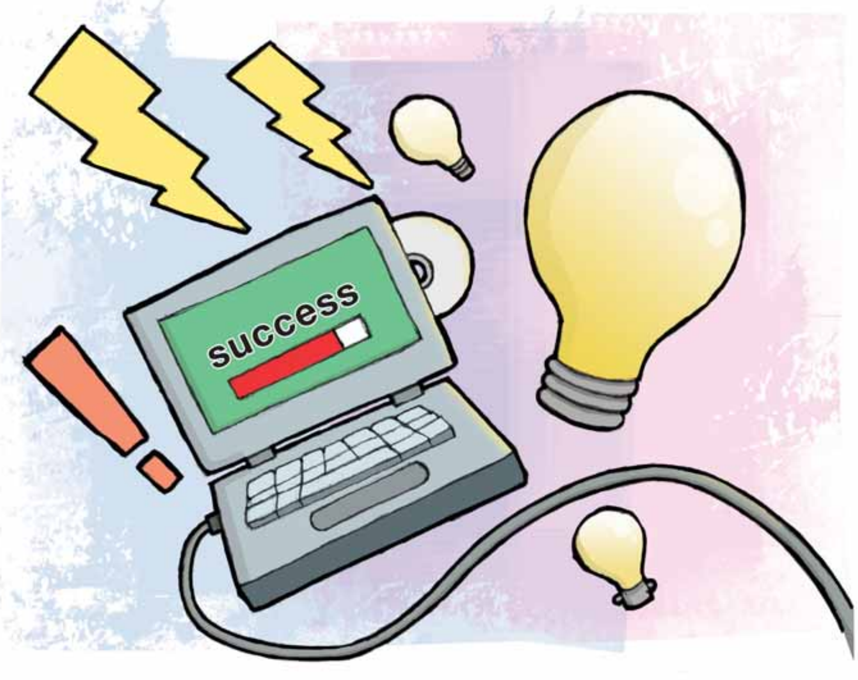
Скорая помощь

Во время изучения пригодится любая информация, а журналы – в особенности. Откройте **config/boot.php** и найдите в нем ak_define('LOG_EVENTS', False);. Замените false на true. Теперь, откройте оболочку командной строки и отобразите последние строчки файла журнала командой

```
tail -f 127.0.0.1
```

Вы увидите запросы, SQL-код и многое другое – очень хорошо для отладки.

Cron: твой ПК



Ежедневное задание *Cron* поможет вам выполнить работу, отдохнуть и поиграть. **Д-р Крис Браун** покажет, как настроить crontab'ы и применить *Anacron*, чтобы ваша система работала, как часы.

Как у вас с греческим? Вы знаете, что *Χρόνος* [chronos, хронос] означает «время»? Конечно, знаете! А в греческой мифологии Хронос был богом и воплощением времени. Поэтому у нас есть такие слова, как хронометр и хронология. Программисты никогда не славились своим правописанием, поэтому сервис Linux, который запускает определенные команды в указанное время, назван Cron.

На нашем уроке вы изучите, как настроить Cron, чтобы планировать собственные задачи, и разберетесь с настройкой файла конфигурации Cron, поставляемого с вашим дистрибутивом Linux. Cron превосходен для машин, запущенных на постоянную работу, но во второй части урока мы рассмотрим его младшего братца Anacron, более подходящего для персональных компьютеров, которые часть времени проводят выключенными.

Часть 1: Осваиваем Cron



Наш эксперт

Д-р Крис Браун внештатный консультант по Linux со степенью доктора наук по физике элементарных частиц и обладатель сертификатов Novell CLP и Red Hat RHCE. Он только что написал книгу по SUSE Linux для O'Reilly.

Cron существует уже довольно давно (хотя и не так давно, как Хронос). Даже в моем потрепанном руководстве шестой редакции Unix, датированном 1974 г., есть упоминание о нем. С тех пор он несколько раз переписывался, и версия, которая в настоящий момент присутствует в Linux-дистрибутивах обычно известна как *Vixie Cron*, написанная Полом Викси [Paul Vixie].

Проще говоря, *Cron* – это демон, запускающий определенные команды в определенное время дня и/или определенные дни недели. Он ежеминутно просыпается и просматривает свои файлы настройки – известные как crontab'ы [таблицы cron] – в поисках назначенных заданий на текущую минуту. Если задание найдено, оно выполняется. Затем программа снова на минуту засыпает. *Cron* обычно стартует во время загрузки.

Где пригодится Cron

- Смена файлов журналов (более подробно об этом см. в статье *«Расчистка файлов журналов»* в LXF92).
- Анализ файлов журналов средствами вроде Logwatch и Webalizer.
- Автоматическая проверка доступности обновлений в репозиториях.
- Перестройка базы slocate. (Эта база всех имен файлов в системе, используется для поддержки команды slocate.)
- Создание резервных копий файловой системы (просто не забудьте заправить ленту в привод перед уходом домой!).
- Стирание устаревших файлов из каталогов типа **/tmp**.
- Запуск систем обнаружения вторжений типа Tripwire.
- Запуск «в ночное» перекомпиляции программ.

Мы используем *Cron* для планирования различных заданий; некоторые примеры приведены во врезке *Где пригодится Cron*, ниже. Большинство дистрибутивов Linux поставляют *Cron* уже настроенным на выполнение некоторых задач, но добавить свои вовсе не сложно, если знаешь как. Люди проделывают с *Cron* невероятные вещи. Мой коллега однажды написал скрипт, собирающий статистику о занятости машины. Используя *Cron*, он запускал скрипт каждые 0, 15, 30 и 45 минут от начала каждого часа и отправлял результаты на центральную машину. Другой скрипт *Cron* запускал каждые 5, 20, 35 и 50 минут от начала каждого часа на центральной машине, чтоб обработать статистику, а затем предоставить ее в виде HTML-отчетов для инженеров и менеджеров. Он также создавал графики нагрузки на машины в зависимости от времени дня.

Другой мой коллега использовал *Cron* для скачивания спутниковых изображений из бюро погоды; он делал это пару лет, а затем свел их вместе и получил кино. Я знаю менеджера, который при помощи *Cron* отключал права на запуск всех игр в 9:00 каждое утро и затем восстанавливал их вечером; а один родитель употребил его для блокировки входа в Интернет (с помощью *iptables*) каждый вечер, чтобы дети не лазили по Сети всю ночь, а наутро восстанавливал доступ.

О том, что и когда делать, *Cron* узнает из набора файлов, называемых crontab'ами. Прежде всего, есть **/etc/crontab**, иногда называемый системным crontab'ом. Анатомия одной записи в этом файле показана на диаграмме напротив. В каждое из пяти временн*ы*х полей вы можете поместить:

- *, означающую любое значение.
- Одиночное значение.
- Список значений, разделенных запятыми, например, 15, 25, 40.

на автомате

» Диапазон, например 9-18.

» Период; обычно следует за *. Например, */5 в поле минуты будет означать «каждые пять минут».

Вот несколько примеров:

```
*/10 * * 5,7,11 * root command1
0 9-18 * * 1-5 root command2
```

Первая строка показывает, что command1 будет запускаться каждые 10 минут в мае, июле и ноябре, а command2 – раз в час во время рабочего дня (с 9 до 18) с понедельника по пятницу. Видите, какая гибкая система! Между прочим, шестое поле в записи crontab, где в наших примерах стоит root, определяет идентификатор пользователя, под которым будет запущена программа. Вы почти всегда будете видеть root в этом поле, пока осторожный администратор не воспользуется принципом наименьших привилегий и не выберет пользователя с минимально необходимыми для запуска программы правами.

Хотя вполне дозволено запускать программы прямо из crontab, сейчас в Linux-дистрибутивах пошла мода на промежуточный скрипт, обычно называемый *run-parts*. Например, вот системный crontab из Red Hat и Fedora:

```
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Скрипт *run-parts* довольно прост: он тупо запускает все исполняемые файлы в заданном каталоге. Есть, как видите, четыре таких каталога, для разных частот запуска. Эта схема упрощает вам жизнь, поскольку надо всего лишь скопировать новый скрипт в соответствующий каталог (в зависимости от того, как часто вы хотите его запускать), и *Cron* автоматически позаботится о его запуске; но вы теряете гибкость настройки времени выполнения, и все скрипты будут выполняться от имени root. И, конечно, такая схема не способна поддерживать операции вроде «запретить игры утром и разрешить вечером», как в примере ранее.

По виду **/etc/crontab** в OpenSUSE 10.2 можно предположить, что его авторы почти полностью забросили системный crontab. Там есть только одна строка, которая выполняет скрипт **/usr/lib/cron/run-crons** каждые 15 минут. Этот недокументированный и непостижимый среднему уму сценарий, судя по всему, запускает скрипты в каталогах **cron.hourly, cron.daily, cron.weekly, cron.monthly**, хотя его логика темна.

Cron для простых смертных

В дополнение к общесистемной таблице **/etc/crontab**, *Vixie Cron* позволяет каждому пользователю вести свою таблицу. Они хранятся под именами пользователей в **/var/spool/cron/crontab** (точное расположение зависит от конкретного дистрибутива).

Эти файлы имеют тот же формат, что и общесистемная таблица, за одним исключением: там нет поля идентификатора пользователя, так как пользователи запускают эти задачи только сами. Редактировать эти файлы напрямую нельзя; для этого надо запустить **crontab -e**, которая позволит открыть файл в определенном вами редакторе (вообще-то редактор определяется переменной окружения **EDITOR**). Зачем нужна оболочка **crontab -e**? Во-первых, она проверяет синтаксис новых записей, прежде чем их установить, а во-вторых, изменяет метку времени на каталоге spool, сообщая *Cron*, что один из файлов был изменен. (Что необычно для демона Linux – не надо извещать *Cron* об изменении файла настроек: он автоматически заметит подмену при пробуждении.)

Администраторы многопользовательских систем могут контролировать, кому из пользователей разрешено использовать *Cron*, размещая их имена в одном из двух файлов, **/etc/cron.allow** и **/etc/cron.deny**. Пример *Может ли Фред запускать Cron?* на следующей странице показывает логику принятия решения. Данная возможность, вероятно, была весьма востребована во времена машин с несколькими тысячами учетных записей; но менее актуальна сейчас, когда машины используются одним-двумя пользователями.

Заоблачный проект

Один из аспектов *Cron*, который пока игнорировался – окружение, в котором выполняются скрипты. Задания обычного пользователя *Cron* работают в его домашнем каталоге. Число переменных окружения является минимальным. *Cron* не считывает скрипт входа пользователя для установки настроек; в частности, в путях поиска находятся только **/usr/bin** и **/bin**, поэтому будьте внимательны, запуская программу из crontab – собственно говоря, общепринятой практикой является использование полных путей команд.

Также принято перенаправлять стандартный вывод и канал ошибок в файл, поэтому вы часто можете видеть в *Cron* подобные записи:

```
15 6 * * * /usr/bin/
someprog > /tmp/
someprog.out 2>&1
```

Если задание *Cron* производит вывод, который не был перенаправлен, то он будет послан по почте владельцу задачи.

При желании вы можете определить собственные переменные окружения в вашей cron-таблице. Например, можно определить командный интерпретатор и расширить путь поиска с помощью таких строк:

> «Cron и Anacron запускают команды в заданное время.»

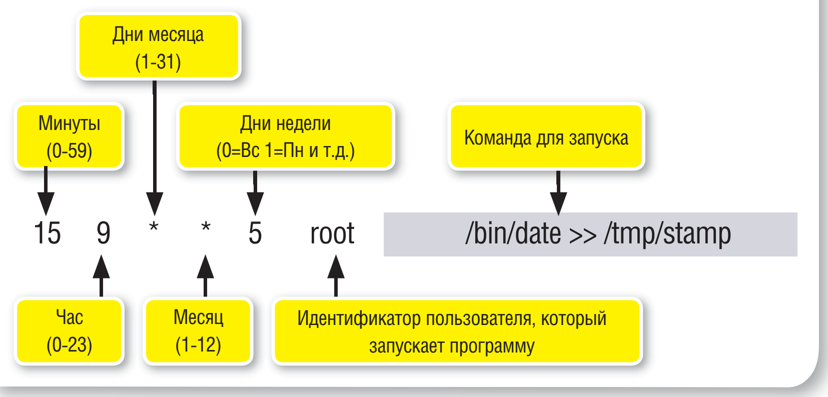
```
SHELL=/bin/bash
PATH=/usr/bin:/bin:/usr/local/bin
```

Если вы, как и я, учитесь, экспериментируя, то, возможно, уже решили сами сделать пару записей в crontab, но если кому нужны мои предложения, попробуйте вот что:

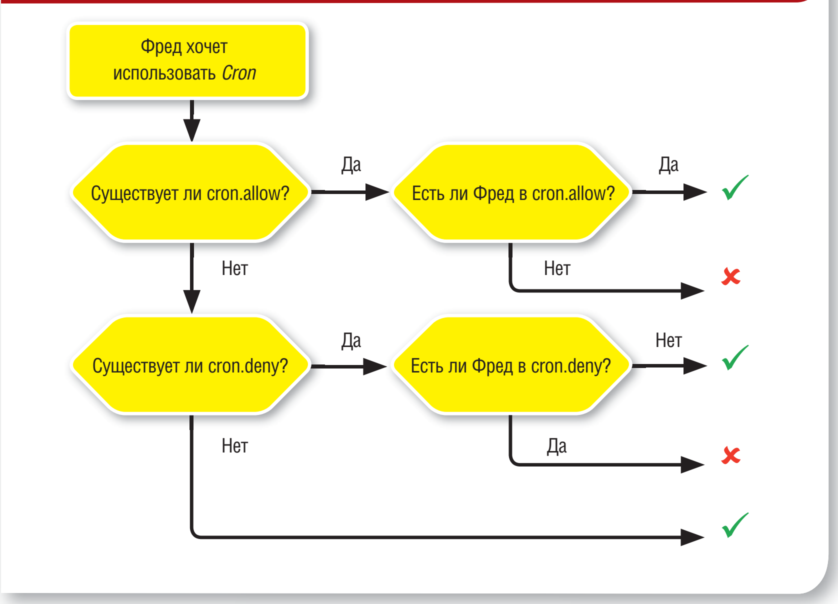
1 Запустите **crontab -e** и поместите в ваш crontab следующие строки:

»

Анатомия записи Cron



Может ли Фред запускать Cron?




```
FOO=BAR
PATH=/usr/bin:/bin:/usr/chris/bin
*/5 * * * * env >> env.out
*/10 * * * * echo "hello from a cron job"
30 * * * * wget http://metoffice.gov.uk/satpics/latest_ir.jpg
```

2 Проверьте, что ваш crontab правильно установлен, командой **crontab -i**.

3 Займитесь чем-нибудь до конца часа. Может, дочитаете журнал?

4 В конце часа проверьте, что в вашем домашнем каталоге появился файл **env.out**. Он должен содержать множество идентичных копий программного окружения *Cron*, предваренных отметками времени через пятиминутные интервалы. Вы нашли переменные окружения, которые сами определили в crontab?

5 Почитайте свою почту (я имею в виду не почтовый POP-ящик из *Evolution*, а ваш локальный почтовый файл, который можно прочесть командой *mail*). Видите сообщения от задачи *Cron*? Они должны появляться каждые 10 минут.

6 Вы получили инфракрасное изображение с метеоспутника? (при необходимости *wget* автоматически добавит номер к файлу, чтобы избежать его перезаписи, поэтому у вас могут встретиться имена **latest_ir.jpg.1, latest_ir.jpg.2** и т.д.). Убедитесь, что Англия покрыта толстым слоем облаков. Если нет – вам, видимо, случайно достался снимок другой страны.

7 Если вы не хотите держать этот crontab дальше, удалите его с помощью **crontab -r**.

Первая запись в crontab в нашем эксперименте просто сохраняет среду, в которой будет выполняться задание. Вторая запись показывает, что если задача создаст вывод, не перенаправленный в файл, то он будет послан пользователю по почте. А третья запись показывает как получать инфракрасное изображение со спутника каждый час. Создание ролика из картинок будет вашим домашним заданием (как принято говорить во всех приличных учебниках).

Скорая помощь

Не пытайтесь использовать *Cron*, чтобы выполнить задачи в 3 утра, если ваш компьютер выключен всю ночь – они не будут выполнены. Вместо этого воспользуйтесь *Anacron*.

Часть 2: Эксперимент с Anacron

Cron превосходен, но имеет слабое место: он спроектирован для машин, которые работают долгое время. Если ваша машина отключена на момент запуска задачи, то задача просто не отработает. Поэтому, если вы планируете на ночь выключать компьютер, то штатные задачи *Cron* (в 4:02 на Red Hat/Fedora и в 6:25 на Ubuntu) не выполнятся никогда.

Чтобы справиться с такими ситуациями, был создан новый инструмент *Anacron*. Согласно лингвистике, «анахроничный» означает «неуместный по хронологии»: например, документальный фильм, демонстрирующий Льва Толстого за ноутбуком, вызовет разве что смешки. Так или иначе, миссия *Anacron* состоит в обеспечении отработки обязательных ежедневных, еженедельных и ежемесячных задач, при условии, что компьютер будет хотя бы раз в день (или в неделю, или в месяц) включен. *Anacron* не интересуется точным временем, назначенным для выполнения задания.

Файл настройки *Anacron*, **/etc/anacrontab**, довольно прост: его синтаксис показан на диаграмме ниже. Вы видите из рисунка, что наименьший период между задачами в *Anacron* – это день; однако *Anacron* не является заменой *Cron*. Вы не можете велеть *Anacron* запустить команды в определенный час и минуту, поэтому (вспомним мой пример с *Cron*) нельзя использовать *Anacron* для отключения доступа в Интернет вечером и восстановления его утром. Чтобы отслеживать, какие программы были запущены и когда, *Anacron* пишет отметки времени в файлах, помещенных в каталоге **/var/spool/anacron**, имена которых соответствуют идентификаторам задач в его таблицах. Отметка времени содержит только день, месяц и год, когда была выполнена последняя задача, но не приводит часы, минуты или секунды.

Анатомия таблицы Anacron



Продвинутый дизайн

Anacron не является демоном в обычном смысле: демоны запускаются при загрузке системы и подстерегают удобный момент, чтобы что-то проделать, а *Anacron* при каждом запуске ищет, что надо выполнить, затем выходит. Резонно спросить: и когда же он в самом деле запустится? Пристальный взгляд на файлы настройки Red Hat Enterprise Linux (или Fedora) показывает, что *Anacron* запускается во время загрузки, через скрипт **/etc/init.d/anacron**. В этот момент *Anacron* считывает свою таблицу, проверяет отметки времени файлов в **/var/spool/anacron**, чтобы узнать, когда каждая задача работала в последний раз, а затем запускает задачи, подлежащие выполнению.

Для каждой задачи, определенной в таблице anacrontab, выждав определенное время, *Anacron* запустит команду. Задержки предназначены для того, чтобы *Anacron* не пытался выполнить все одним махом после загрузки машины, так как пользователь захочет приступить к

делу, а не ждать, пока система освободится.

Сам по себе *Anacron* не особо сложен в понимании. Однако все становится куда интереснее, если попытаться выявить связь между *Anacron* и *Cron*, особенно если одинаковые задачи выполняются в обеих программах. В настройках Red Hat/Fedora таблица *Anacron* выглядит так:

```
1 65 cron.daily run-parts /etc/cron.daily
7 70 cron.weekly run-parts /etc/cron.weekly
30 75 cron.monthly run-parts /etc/cron.monthly
```

Из этой выдержки мы видим, что *Anacron* просят выполнять в точности те же задачи, что и *Cron*, то есть в соответствующие интервалы (1, 7 или 30 дней) он использует скрипт *run-parts* для запуска всех скриптов в **/etc/cron.daily, /etc/cron.weekly** или **/etc/cron.monthly**. Вы заметите, что *Anacron* не запускает скрипты *cron.hourly*. Как я уже сказал, он для этого не предназначен, потому что записывает отметки времени лишь с точностью до дня.

Сойдите с моей ноги

Когда одинаковые задачи выполняются и *Cron*, и *Anacron*, возникает опасность того, что *Cron* запустит задачу, которую уже выполнил *Anacron*, или наоборот. Разные дистрибутивы используют разные трюки для предотвращения этой ситуации. Если внимательно рассмотреть настройку Red Hat\Fedora, то можно обнаружить, что каталог **cron.daily** содержит скрипт с именем *0anacron*, запускающий *Anacron* с опцией **-u**. Так, мы видим, что *Cron* запускает *Anacron* каждый день. Однако опция **-u** просто говорит *Anacron* обновить отметки времени, без запуска самих программ. Фактически, *Cron* сообщает *Anacron*: «Пожалуйста, учти, что эти команды я уже выполнил сегодня (или на этой неделе, или в этом месяце...), и не трогай их».

Ubuntu использует другой прием. Здесь таблица cron выглядит следующим образом:

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
17 * * * * root run-parts /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || run-parts /etc/cron.daily
47 6 * * 7 root test -x /usr/sbin/anacron || run-parts /etc/cron.weekly
52 6 1 * * root test -x /usr/sbin/anacron || run-parts /etc/cron.monthly
```

Часть **hourly** выглядит понятно; она более-менее похожа на Fedora. Записи на день, неделю и месяц используют трюк с командным интерпретатором, чтобы удостовериться, что *run-parts* запускается только в отсутствии *Anacron*. (См. *Магический трюк с run-parts* во врезке вверху справа.) Обобщая разницу в подходах Red Hat и Ubuntu, можно сказать: «В Red Hat выигрывает *Cron*. В Ubuntu победа за *Anacron*.»

Настройки по умолчанию для Cron и Anacron в различных дистрибутивах могут запутать, однако я бы сказал следующее: вы не обязаны использовать *Cron* и *Anacron* именно так, как определил ваш поставщик Linux. Если вы понимаете работу этих двух сервисов и синтаксис файлов настройки, можете заставить их делать то, что хотите вы.

Магический трюк с run-parts

Оператор командного интерепретатора ИЛИ, ||, соединяет два логических выражения, чтобы вычислить результат в виде **true/false**. Например a || b = **true**, если a дает **true** или b дает **true**. (То есть, если команда a или команда b возвращают 0 при выходе.) Теперь – внимание, здесь вся соль! – если a возвращает **true**, значение a || b будет **true** независимо от значения b. Ясно? И в этом случае команду b даже незачем выполнять. Это довольно простое поведение интерпретатора как раз и объясняет то, что мы видим в таблице cron в Ubuntu. Команда интерпретатора

```
test -x /usr/sbin/anacron || run-parts /etc/
cron.daily
```

запустит две команды, *test* и (возможно) *run-parts*. Команда test встроена в интерпретатор; test -x возвращает **true**, если определенный файл существует и является исполняемым. Поэтому это строка интерпретируется так: «Если существует **/usr/sbin/anacron**, и он является исполняемым, тогда все выражение равняется **true**, и нет нужны запускать *run-parts*, чтобы это определить», или, по-другому, «запускай *run-parts,* только если *Anacron* не присутствует». Quod erat demonstrandum («Что и требовалось доказать»), как говаривали древние римляне.

Скорая помощь

Не думайте, что вы должны придерживаться правил, заданных вашим дистрибутивом. Как только вы поняли формат этих файлов, то сможете заставить выполнить Cron что угодно и когда угодно.

Часть 3: Одноразовые задачи с at

У нас как раз осталось место, чтобы рассказать о команде *at*. В отличие от *Cron* и *Anacron*, обеспечивающих повторяемые действия, *at* используется для планирования разовых действий в будущем. Положим, вы захотите, чтобы ваш компьютер пересобрал ядро Linux, пока вы проводите вечер в баре. Вы можете сделать это таким образом:

```
$ at 20.30
warning: commands will be executed using /bin/sh
at> cd /usr/src/linux
at> make all
at> <EOT>
job 2 at Wed Feb 14 19:30:00 2007
```

Как видите, at получает команды, которые вы хотите выполнить, через стандартный ввод до ожидания символа **EOT** (по умолчанию, ^D). Ваш список поставленных задач можно посмотреть через команду *atq*.

```
$ atq
2 Wed Feb 14 19:30:00 2007 a chris
$
```

Демон *at* (*atd*) запустит задачу в назначенный момент, экономя вам время на более интересные занятия. Правда, *at* не доставит вас в бар в назначенное время и не предложит напитков. Однако вы можете из вежливости купить ему пакетик чипсов.

Подробности об очереди задач хранятся в специальном каталоге (обычно **/var/spool/at**). Если вы определите задачу или две, а затем посмотрите содержимое этого каталога, то заметите, что в отличие от *Cron*, *at* трудится над созданием такой же среды выполнения, как если бы вы только что зашли. Демон *atd* ответственен за реальное выполнение задач.

Если вы вдруг передумаете и решите отменить задачу *at*, можете сделать это командой *atrm*:

```
$ atrm 2
$
```

Аргументом для *atrm* является номер задачи (в данном примере это 2).

Больше всего меня поражает в этой программе изобилие форм записи для задания времени выполнения команды. Верны все следующие варианты:

```
$ at 14.30 tomorrow
$ at noon Friday
$ at 4pm + 3 days
```

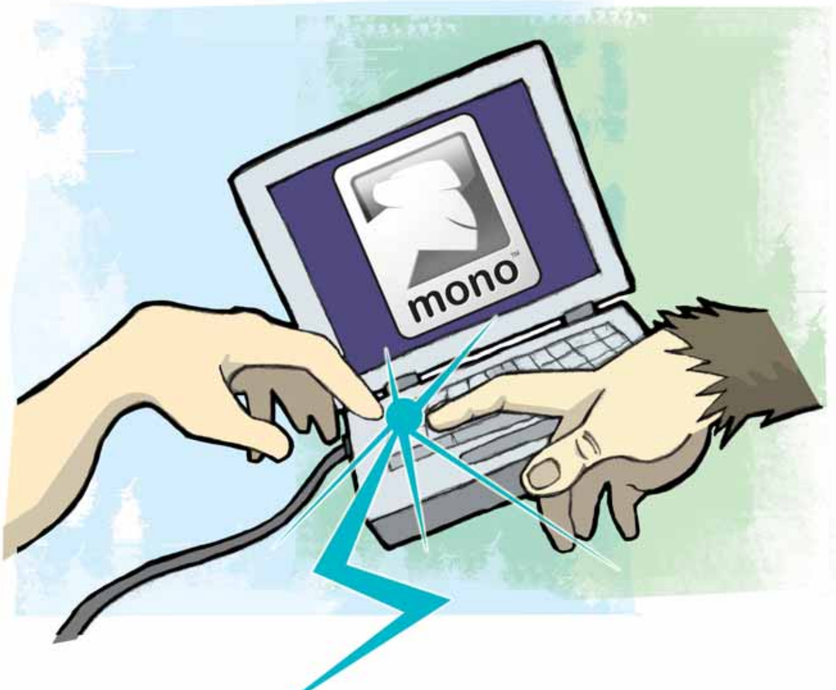
Моим любимцем является ключевое слово teatime как синоним 4 часов дня:

```
$ at teatime tomorrow
```

Красота!

Mono: Легкий

Едва вы начинаете делать что-нибудь интересное, XML сразу же становится медленным и громоздким, и **Пол Хадсон** думает, что пора взяться за работу с базами данных...





Наш эксперт

Пол Хадсон полагает, что Mono – лучшая вещь со времен мультфильма Pinky and the Brain, и сейчас поддерживает два проекта на основе Mono на SourceForge.

Поскольку это — последний номер, который попадет к редактору Ребекке, скажу не таясь: пишет она уж больно мелко. Когда мы возвращаемся из отпуска, на наших столах нас ждут ковры из желтых листков-наклеек для заметок, покрытых неразборчивыми каракулями. На вид сущая грязь, пока вы не вооружитесь лупой и не разглядите ее аккуратный почерк. Конечно, потом начинается надсадная для глаз работа – читать все эти записки, а ведь можно было обойтись всего лишь SQL-запросом, в стиле SELECT Значимое FROM ЗапискиРебекки WHERE Приоритетность!=Не относится к делу. Увы, все попытки заставить Ребекку взаимодействовать через базу данных потерпели крах, и теперь она ищет другую работу. Но вы-то все еще здесь, верно? Значит, у меня есть время, чтобы подсадить вас на SQL, прежде чем вы заскучаете от Mono и перейдете к разделу Hardcore Linux!

На этом уроке мы напишем небольшой инструмент, способный следить за всеми Mono-программами, которые вы создали на протяжении всей нашей серии. Каждые десять секунд программа будет опрашивать Mono о списке всех активных приложений, а затем сохранять этот список в базе данных. Мы получим шанс изучить два новых аспекта Mono: чтение информации о процессах и доступ к базе данных. Этот проект – еще одна звездочка вам на погоны! Как и для всех наших Mono-проектов, потребуется среда исполнения .NET 2.0; убедитесь, что вы ее включили, через диалог Project > Options > Runtime Options.

Чтение процессов

Начните новый консольный проект в MonoDevelop и дайте ему имя Monocular. Первым делом надо прочесть список всех процессов, работающих в Mono, и вывести информацию на экран. Решить эту задачу довольно просто, благодаря пространству имен System.Diagnostics, которое дает доступ к россыпям внутренней системной информации. Нас интересует класс Process, позволяющий читать информацию о запущенной программе. Помните, что «программа» – это исполняемый код на диске (например, *Vim*), а «процесс» – экземпляр программы, который в данный момент работает. То есть если вы запустите *Vim* шесть раз, в оперативной памяти будет шесть процессов *Vim*, но программ от этого не прибавится.

Добавьте using System.Diagnostics в начале файла **Main.cs**, затем замените стандартную строку Hello World на следующее:

```
Process[] proclist = Process.GetProcesses();
foreach(Process proc in proclist) {
  Console.WriteLine(proc.ProcessName);
}
```

Для работы этого достаточно, поэтому нажимайте F5 и смотрите на вывод панели Application Output. Вы увидите что-то вроде 'Monocular, MonoDevelop, mdhost, mdhost, MonoDevelop…', а также все прочие запущенные Mono-программы – Beagle, Tomboy, F-Spot и так далее. Если у вас сравнительно новая версия MonoDevelop (Fedora Core 6 или выше), будет доступно автоматическое завершение кода – вы это заметите, если нажмете . (точку) после proc: появится обширный набор свойств, которые можно узнать о процессе. Как именно поддерживаются эти свойства, зависит от версии Mono – полюбопытствуйте, что предоставляет ваша!

Внимание: поддержка процессов улучшается с каждым релизом Mono – последние дистрибутивы по умолчанию включают Mono 1.2.3, а моя версия, из Fedora Core 6, уже приотстала.

Знамение времени

Следующая часть нашего проекта – задействовать таймеры, так как нам нужно выполнять периодическую проверку запущенных процессов. Можете написать следующий код:

```
while(1){
  if (прошло достаточно времени){
    делай_работу();
  }
}
```

...но это капитальная глупость: ваше приложение будет тупо съедать процессорное время без особой пользы. Гораздо лучшее решение – применить таймер: он позволит вашему приложению уснуть на заданное время, а затем вызовет определенный вами метод.

Метод, который мы будем использовать, называется TimesUp(). Чтобы мы смогли подключить его к таймеру, он должен принимать определенный список параметров. Создайте следующий метод:

```
static void TimesUp(object sender, ElapsedEventArgs e){

}
```

Теперь перенесите код из Main() в TimesUp(), потому что мы хотим вызывать его, когда наш таймер сработает. В начале файла напишите следующие строки using:

» **Месяц назад** Взлом хэшей на нескольких ядрах: многопоточность изнутри.



Доступ к данным

```
using System.Threading;
using System.Timers;
```

Чтобы ваша программа вызывала код каждые 10 секунд, создайте новый таймер и укажите ему метод TimesUp():

```
System.Timers.Timer timer = new System.Timers.Timer();
timer.Interval = 10000;
timer.Enabled = true;
timer.Elapsed += new ElapsedEventHandler(TimesUp);
```

Не пытайтесь запустить код в MonoDevelop! Когда вы работаете с потоками, а с таймером так дело и обстоит, MonoDevelop обычно некорректно останавливает программу в режиме отладки; поэтому лучше запустить ее из консоли.

При запуске не произойдет никакого вывода на экран. С чего бы это? Мы возвращаемся к проблеме с потоками, рассмотренной в прошлый раз: программа достигает конца Main() и завершает работу до того, как таймер сработает. Решение состоит в приказании главному потоку (он-то и исполняет Main()) уснуть, посредством метода Thread.Sleep(). Метод позволяет задать число миллисекунд, на которые поток впадет в спячку, или же указать Timeout.Infinite, если вы хотите, чтобы поток уснул навеки. Последняя опция нам и нужна – пусть главный поток спит, а таймер будет вкалывать.

В конце Main() добавьте строку:

```
Thread.Sleep(Timeout.Infinite);
```

Теперь при запуске программы она будет честно раз в 10 секунд выводить список процессов. Прежде чем продолжать, сделаем еще кое-что: добавьте такую строку перед вызовом Thread.Sleep():

```
TimesUp(null, null);
```

Предоставляю вашему воображению – и тестированию! – догадаться, что она делает.

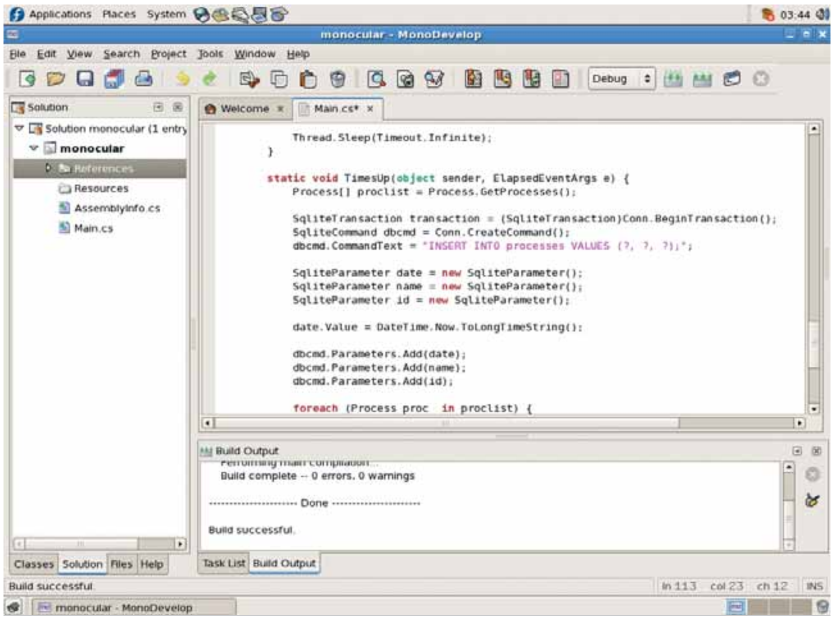
Создание базы данных в Mono

Информацию о процессах мы получили; надо ее куда-то положить. Базы данных бывают двух типов: клиент-серверные, когда данные хранятся где-то на сервере; и встроенные, когда есть локальный файл, напрямую управляемый из программы. Клиент-серверных баз данных полным-полно – MySQL, PostgreSQL, Oracle и другие соревнуются за место под солнцем. Но среди встроенных, одна из баз заслуживает большего внимания – это *SQLite*. *SQLite* хороша по нескольким причинам: ее код является достоянием общественности и вы можете делать с ним все, что хотите; у нее есть продвинутые возможности – поддержка Unicode, транзакций и, конечно, SQL-запросов; она также очень быстрая. Используя *SQLite* в Mono, можно считывать и записывать файл базы данных прямо из C#.

Слишком хорошо для этого мира? Да нет, загвоздка имеется: с базами данных не так легко работать через Mono. Если вы следовали инструкциям по установке в нашем первом выпуске (LXF87/88), то уже разрешили зависимости Mono для *SQLite*; если нет, то я включил DLL-файл на диск, и вы можете его скопировать в папку с программой. В любом случае надо сделать ссылку на Mono.Data.SqliteClient. Есть небольшая разница между версией, входящей в Mono, и DLL – я постарался ее минимизировать, но если вы используете DLL, то вам понадобятся большие буквы в некоторых именах классов (т.е. SQLiteConnection вместо SqliteConnection).

В любом случае, *SQLite* использует для хранения данных обычный файл, так что первым шагом будет его создание. Добавьте следующие строки в начало вашего файла:

```
using Mono.Data.SqliteClient;
using System.IO;
```



› *MonoDevelop*: правильная среда для компиляции проекта, но не для запуска – из-за проблем с потоками. Используйте командную строку!

Если вы используете DLL и обнаружили, что Mono.Data.SqliteClient не существует, попытаетесь набрать Mono.Data. и выбрать верное название сборки из предложенного списка опций.

Создание базы данных осуществляется с помощью метода File.Create(), но фактически базу надо создать только один раз – если файл уже существует, незачем пересоздавать его. Поэтому предусмотрим проверку: первый раз запускается программа или нет, а также заведем место для хранения информации о соединении с нашей базой данных *SQLite*. Добавьте следующие две строки под class Monocular {:

```
static bool FirstTime = false;
static SqliteConnection Conn;
```

Теперь подправим Main(), чтобы он автоматически создавал базу при первом запуске приложения, а затем спокойно к ней подсоединялся:

```
if (!File.Exists("monocular.db")){
 Firsttime=true;
 File.Create("monocular.db");
}
Conn =new SqliteConnection("URI=file:monocular.db");
Conn.Open();
```

Итак, файл для хранения ваших данных создан. Чтобы *SQLite* могла писать в него данные, надо еще создать таблицы, определяющие сохраняемые поля. Все команды *SQLite* выполняются через объекты SqliteCommand. Новая непонятка? А вы продолжайте читать. Эти SqliteCommand-объекты уникальны в Mono: несмотря на потуги программистов – разработчиков .NET-коннектора, они не убирают за собой. Если в Mono вы выделяете память под объект, то вам не надо заботиться о ее освобождении – сборщик мусора Mono автоматически сделает это за вас. *SQLite* иногда выделяет память, которая впоследствии автоматически не освобождается, и вам придется помнить об этом самим.

В данный момент вам надо создать объект типа команда, настроив его таким образом, чтоб он создавал таблицу для хранения информации о процессах; затем выполнить команду. Вот как это выглядит на C#:

```
   SqliteCommand dbcmd = Conn.CreateCommand();
   if (FirstTime) {
```

»

```
    dbcmd.CommandText = "CREATE TABLE processes
  (DateStamp TEXT, Name TEXT, Id INTEGER);";
    dbcmd.ExecuteNonQuery();
  }
```

Если вы изучали SQL раньше, то удивитесь, почему поля отмечены как TEXT и INTEGER, а не более научно, как VARCHAR(255). Дело в том, что *SQLite* не содержит таких четких определений. Поля хранятся либо как числа, либо как текст. Если же вы не изучали SQL, то кусок кода выше создает таблицу для хранения полей DateStamp, Name и Id – мы будем использовать их для хранения даты каждого чтения, имени запущенного процесса и его идентификатора.

Вообще-то ExecuteNonQuery() – не самое умное имя: Q в аббревиатуре SQL означает Query – Запрос, то есть каждое предложение SQL является запросом. Ну, а ExecuteNonQuery() означает, что вы хотите заставить *SQLite* сделать что-то, но результаты запроса вас не интересуют.

После создания таблицы необходимо очистить память, выделенную командой, следующим образом:

```
dbcmd.Dispose();
dbcmd=null;
```

Вы, возможно, удивитесь, почему эти две строки кода не входят в блок if – а потому, что мы попозже добавим еще код!

В любом случае, хотя код компилируется и запускается, проку от него мало – ничего интересного он не делает.

Скорая помощь

SQLite – не единственная база, доступная из Mono: есть еще *MySQL*, *PostgreSQL* и другие, к которым есть сопутствующие библиотеки.

Время писать!

OK, у вас есть таймер, у вас есть список процессов, а теперь еще и доступ к базе данных – налицо весь инструментарий для создания всего проекта. Единственное, чего мы не умеем –писать в базу данных, но тут имеются осложнения. Противно то, что дело это нехитрое, но в наш отдел писем недавно поступила жалоба на мой «непонятный» стиль изложения. А я решил привести оптимальный способ создания записей в базе данных – оптимальный с точки зрения производительности, а не с точки зрения обучения. Кто не согласен, пишите мне, и мы сменим акценты.

Так, отвел душу – и хватит; вернемся к базе данных! Есть две вещи, которые необходимо знать, прежде чем приступать к записи в базу *SQLite*:

1. *SQLite* использует транзакции для обеспечения целостности данных. То есть, когда вы пишете строку данных, то запускается транзакция, записываются данные, и транзакция завершается. Если вдруг ваша машина откажет во время записи транзакции, то SQLite ее откатит. Это очень хорошо.
2. *SQLite* кэширует запросы для ускорения работы.

Первый пункт очень важен, так как мы собираемся писать за раз сразу несколько строк. Если у вас работает 10 Mono-приложений, будем писать 10 строк – по одной для каждой программы. Процесс в *SQLite* будет выглядеть следующим образом:

1. Начать транзакцию
2. Записать строку
3. Закончить строку
4. Начать транзакцию
5. Записать строку
6. Закончить транзакцию… и так далее.

Скорая помощь

Проверьте вашу версию Mono – в новых релизах есть методы ToBinary() и FromBinary() для DateTime.Now, они гораздо лучше для хранения дат. Использование строк является простейшим решением, но их не отсортируешь – примените целые числа, если сможете.

Это очень медленно, но необходимо, так как *SQLite* требует, чтобы каждая команда записи была частью транзакции. Той же цели можно достичь, если велеть *SQLite* начать транзакцию, записать все строки сразу же, затем закончить транзакцию самим:

1. Начать транзакцию
2. Записать строку
3. Записать строку
4. Записать строку
5. ...
6. Закончить транзакцию.

Такой способ работает много быстрее, и он также более надежен: если наш компьютер откажет (или же другое несчастье прервет работу *SQLite*), то у нас не будет недоделанных записей – у нас будут либо все записи, либо ничего.

Второй пункт важен потому, что подобные запросы очень просто писать:

```
for (int i = 0; i < 10; i++){
$query = "SELECT Foo FROM Bar WHERE Baz = "+i;
...
{
```

Выполнится 10 различных запросов к *SQLite*. Лучше использовать приготовленные заранее запросы и сообщать *SQLite* максимум информации о запросе, помечая заменяемые части вопросительными знаками.

Из примера все будет ясно, поэтому углубимся в код. Пока что TimesUp() у нас распечатывал информацию; а надо заточить его на запись в базу данных. Вот как TimesUp() должен выглядеть:

```
static void TimesUp(object sender, ElapsedEventArgs e) {
Process[] proclist = Process.GetProcesses();
SqliteTransaction transaction = (SqliteTransaction)Conn.
BeginTransaction();
dbcmd = Conn.CreateCommand();
dbcmd.CommandText = "INSERT INTO processes VALUES (?,?, ?);";
SqliteParameter date = new SqliteParameter();
SqliteParameter name = new SqliteParameter();
SqliteParameter id = new SqliteParameter();
date.Value = DateTime.Now.ToLongTimeString();
dbcmd.Parameters.Add(date);
dbcmd.Parameters.Add(name);
dbcmd.Parameters.Add(id);
foreach(Process proc in proclist) {
```

Это еще не вся функция, но я хочу притормозить и взглянуть на подготовленные запросы в действии. Запрос, который мы используем, передает вместо знаков вопроса информацию о процессе, так как только эта часть будет изменяться. Однако параметр date изменяется только один раз за время выполнения TimesUp(), поэтому его можно назначить сразу. Другие параметры просто оставляются как пустые объекты SqliteParameter, чтобы потом заполнять их отдельно для каждого процесса. Знаю, что это тривиально, но ради ясности ситуации скажу: необходимо вызывать dbcmd.Paramters.Add() для каждого параметра SQL-запроса, в порядке их появления в запросе.

Ладно, продолжим с того места, на котором остановились...

```
foreach(Process proc in proclist) {
name.Value = proc.processName;
id.Value = proc.Id;
dbcmd.ExecuteNonQuery();
}
dbcmd.Dispose();
dbcmd = null;

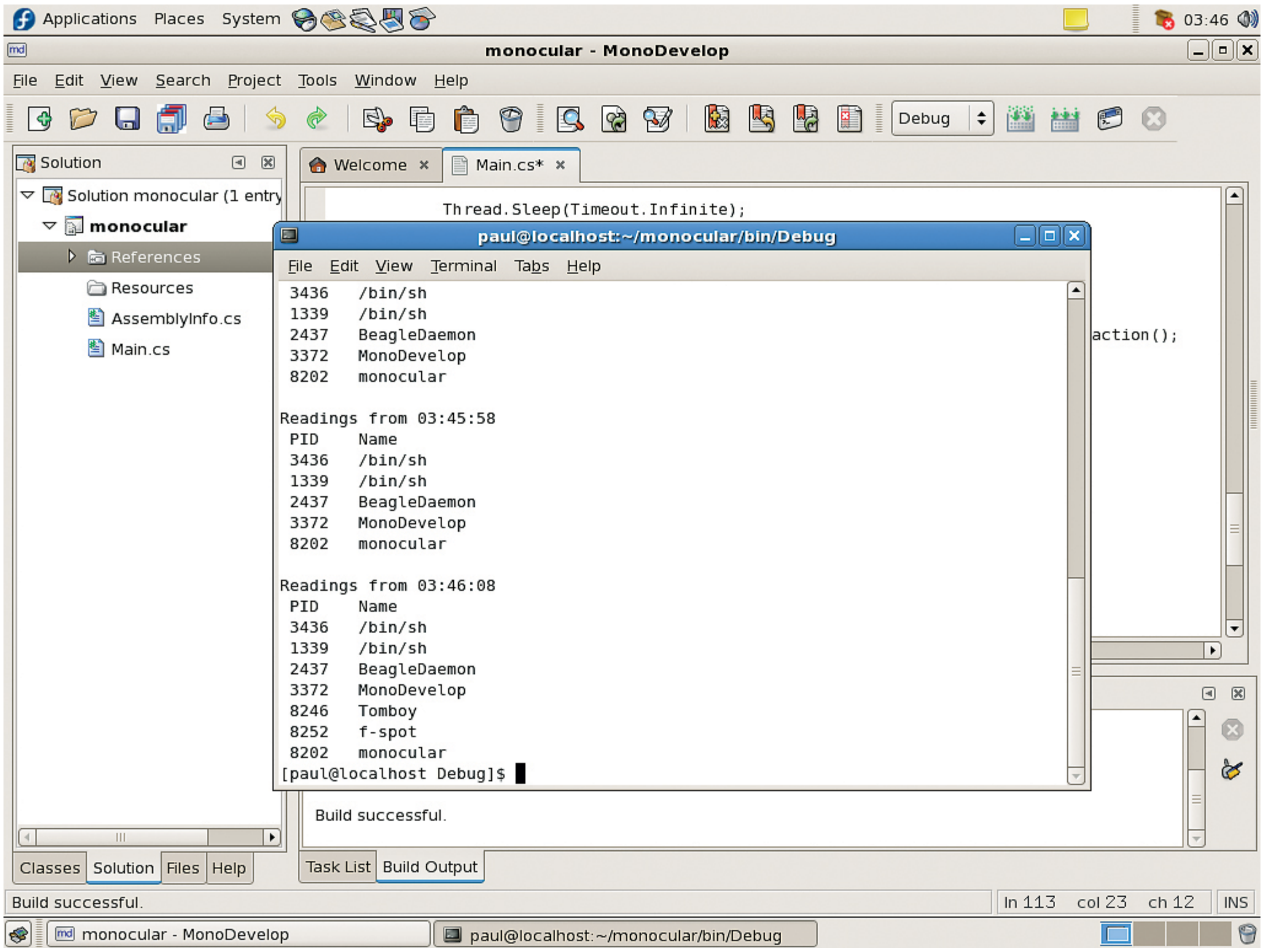
transaction.Commit();
}
```

Действительно, все, что меняется между вызовами ExecuteNonQuery() – это значения двух подставляемых параметров, поэтому SQLite может использовать кэшированные запросы для получения наивысшей производительности. Последнее выражение в коде особенно важно: пока транзакция не зафиксирована, никаких записей

Слабости Mono

Да, я вовсю расхваливал Mono, но теперь вы в состоянии заметить и недостатки – там, где Mono не так хорош, как официальная реализация Microsoft .NET; но это только пока.

Во-первых, при работе в официальной .NET-среде эта программа снабдит вас данными обо всех работающих программах, .NET или других. Во-вторых, официальная среда поставляет информацию о том, сколько памяти использует каждый процесс. В-третьих, официальная среда содержит глобальные счетчики производительности, измеряющие использование CPU, объем свободной глобальной памяти и т.д. Каждый из этих пунктов сделал бы нашу программу куда полезнее, но на беду, Mono не имеет ни одной из перечисленных возможностей – что, конечно, вредит программе этого месяца!



Готовый продукт: ***Monocular*** **следит за работающими Mono-программами, каждые 10 секунд опрашивая состояние системы.**

внесено не будет – не забудьте вызывать Commit(), если, конечно, вы не любитель охоты за странными ошибками! Написано тут много, но смотреть все равно не на что, кроме как на медленное разрастание файла monocular.db. Но скоро все пойдет по-другому...

К чтению готовы!

Последнее, что нам предстоит сделать – это считать всю информацию из базы данных, так что вернемся к методу Main() и проделаем кое-какие изменения. Если пользователь запускает *Monocular* без параметров, то она будет бесконечно работать в фоновом режиме, считывая информацию каждые 10 секунд и записывая ее в базу данных. Но если ей предоставить параметр – любой параметр – *Monocular* будет выводить всю хранимую информацию. На DVD я поместил код, позволяющий выполнять сортировку по имени процессов или по идентификатору, но ради экономии места мы пока проигнорируем его значение. Сейчас нам важен параметр как таковой — он будет сигнализировать о том, что пользователю нужен режим чтения.

В отличие от предыдущих запросов, теперь вам действительно потребуются данные – результаты SQL-запроса; добудем их с помощью специального объекта SQLiteDataReader:

```
string currenttime = “”;
// this loops over all the rows returned by the query
while (reader.Read()) {
string datestamp = reader.GetString(0);
string name = reader.GetString(1);
int id = reader.GetInt32(2);
// we’ll put some more code in here shortly
Console.WriteLine(“ {0}\t{1}\2”, id, name);
}
reader.Close();
reader = null;
}
```

GetString() и GetInt32() оба принимают один параметр: какое поле вы хотите считать. То есть GetString(0) считывает первое поле (DateStamp), GetString(1) считывает второе поле, и так далее. Как и с командами, вам надо на всякий случай освободить память, выделенную для считывателей, так как SQLite может этого не сделать.

Данный код выводит хранимую в таблице информацию обо всех процессах, но игнорирует поле даты. Это потому, что я хочу внести немного разнообразия: буду выводить каждый временной штамп при его изменении, тогда наш вывод будет разделяться чем-то вроде заголовков. Вы заметили пустую переменную currenttime. В нее мы занесем дату первой строки, считываемой из таблицы. Если в последующей строке окажется отличное от currenttime время, то выведется новый заголовок для нового времени, и это значение запишется в currenttime. На C# получится следующее:

```
if (datestamp != currenttime) {
Console.WriteLine(“”);
Console.WriteLine(“Readings from “ + datestamp);
Console.WriteLine(“ PID\tName”);
currenttime = datestamp;
}
```

Вот и все – проект завершен! Я знаю, он не так полезен, как остальные проекты, но у нас связаны руки, так как в Mono отсутствуют некоторые интересные возможности официальной реализации Microsoft. Не горюйте: работа над добавлением недостающих возможностей в Mono ведется постоянно. А пока вы можете много чего еще добавить в эту программу: например, возможность старта и остановки программы, определения периодов наибольшей загрузки системы и так далее. Творите! LXF

Скорая помощь

На смену Fedora Core 6 пришла Fedora 7, а значит, у нас есть возможность поиграть с новой версией Mono. Со следующего выпуска мы будем ориентироваться на Mono 1.2.3. Обновляйтесь!

» Через месяц Пол перейдет на Fedora 7 и напишет чат-клиент на GTK.

DCOP: Моддинг

Нравится идея менеджера виртуальных рабочих столов для KDE? **Грэм Моррисон** покажет, как автоматизировать KDE через функции DCOP.





Наш эксперт

Грэм Моррисон
Разработчик свободных программ, фанат KDE и энтузиаст электронной музыки. Он обсуждал синтезаторы не с одним ведущим членом сообщества FOSS.

У KDE больше функций на каждый пиксель, чем у любого другого рабочего стола, который я в силах припомнить. Это источник гордости и критики для пользователей, которые или обожают, или ненавидят эту бездну настроек. Пользователи KDE привыкли тонко регулировать все аспекты пользовательского интерфейса, как через *KControl*, так и с помощью диалогов настроек каждого приложения. Но большинство пользователей и не подозревают, что KDE предоставляет механизм настроек поинтереснее, чем пара-тройка панелей и вкладок.

Этот механизм можно задействовать через DCOP (Desktop Communication Protocol, Протокол взаимодействия рабочего стола). Его мало используют и плохо понимают, но он способен добраться до самых недр KDE-приложения. И когда я говорю недра, я имею в виду недра: в данном случае, это функции, созданные программистом для построения своего приложения, которые насколько глубоко вникли во KDE, насколько это возможно. На нашем уроке мы создадим скрипт, который поможет вам управлять непомерным количеством окон, обычно висящем на типичном рабочем столе KDE. Наш скрипт перекинет однотипные приложения на один рабочий стол, упростив одновременную работу с несколькими задачами.

DCOP для масс

Вы будете работать с функциями, причем компьютерными. Единственное их сходство с математическими собратьями – то, что они обрабатывают данные и выводят результат. Эти функции являются строительными блоками приложения, в основном используемыми для разбиения больших задач на мелкие фрагменты. Например, вы можете захотеть иметь функцию для свертывания или развертывания окна на рабочем столе. Пользователю незачем знать, какая функция скрывается за данной возможностью: ему нужно только, чтобы при нажатии на определенную иконку происходило определенное действие.

Свертывание и развертывание окон было выбрано не случайно. У всех приложений KDE есть функции для выполнения этих задач, и вот здесь может помочь DCOP [небольшое отступление: функциональность, сходной с DCOP, но не завязанной на KDE, обладает D-BUS – онто и придет на смену DCOP в KDE4, – прим.ред.]. DCOP дает обычным пользователям доступ ко многим «закулисным» функциям, используемым приложениями внутри себя; он позволяет перенаправлять данные в приложение и из него, как будто вы являетесь разработчиком приложения. Конечно, обычные пользователи могут оробеть перед такими возможностями, но раз вы читаете эту статью, значит, вы не просто давнишний пользователь, верно?

DCOP является протоколом межпроцессного взаимодействия. Таких существует довольно много, но от большинства других реализаций DCOP отличает сравнительная доступность. Откройте командную строку, наберите **dcop**, и поймете, что я имею в виду. Вывод данной команды будет содержать список всех KDE-приложений, открытых в данный момент – его любезно предоставляет DCOP.

При запуске каждое приложение присоединяет себя к общесистемному серверу DCOP – его-то вы и опрашивали командой 'dcop'. Но самое интересное начнется, когда вы попытаетесь опрашивать работающие приложения. Например, наберите **dcop kdesktop**. Вы увидите список объектов верхнего уровня, использованных программистами для разработки приложения *KDesktop* – инструмента, ответственного за фон рабочего стола и прочие небольшие задачи. DCOP использует иерархию узлов для упорядочения доступных функций, и то, что вы видите, является списком узлов верхнего уровня присоединенных к *KDesktop*. Вы можете попробовать опросить любое приложение, встретившееся в выводе нашей первой команды *dcop*. Мы на самом деле уже использовали DCOP для манипулирования *KDesktop* пару лет назад в *Совете месяца* в разделе руководств LXF66, когда создавали небольшой скрипт для скачивания космического изображения дня с NASA и автоматической установки его в качестве фонового изображения.

Расшифровка синтаксиса с Kdcop

Такая гибкость является силой DCOP, а его возможности просто безграничны. Наилучший способ исследовать некоторые из этих возможностей – использовать графическую KDE оболочку для команды *dcop*, называемую *Kdcop*. Она редко встречается в системном меню, и вам, вероятно, потребуется запустить ее через пункт Запустить команду из меню KDE и набрать **kdcop**. Это отличная маленькая утилита, позволяющая исследовать различные KDE-приложения, их ассоциированные узлы DCOP и функции.

Пользовательский интерфейс *Kdcop* прост в понимании. Панель поиска наверху особенно полезна, так как обновляется в реальном

» **Месяц назад** Мы создавали интерактивные web-формы с новым тэгом canvas.

в стиле KDE

времени. Попытайтесь набрать 'Wallpaper' в поле поиска – появится несколько записей насчет фона стола, присоединенных к объекту KBackgroundIface.

Функция, которую мы использовали в том *Совете месяца*, называлась void setWallpaper(QString wallpaper, int mode). Если вы никогда не занимались KDE-программированием, вам это мало что скажет – но вообще-то это простой C++ синтаксис для определения входных и выходных параметров [а также возвращаемого значения, – прим. ред.]. Сама функция называется setWallpaper, а значения в скобках – это два параметра, которые она требует на вход: текстовая строка, содержащая путь к изображению, и целое число, обозначающее режим использования. Что за режим? Вот и проявилась самая большая проблема с DCOP: формальной документации нет.

Если вы хотите понять, что делают параметры вроде **mode**, единственный способ это сделать – посмотреть исходный код приложения. Тем, кто использует DCOP с KDE-приложением, не являющимся основной частью рабочего стола, придется получить исходный код с домашней страницы проекта или со страницы SourceForge. К счастью, setWallpaper() является составной частью дистрибутива рабочего стола KDE, то есть вы можете посмотреть исходный код в режиме онлайн (см. **http://techbase.kde.org**) или установить пакеты разработки KDE [другие приложения также позволяют увидеть свой код в Сети, например, все проекты SourceForge предоставляют для этих целей ссылку Browse SVN, – прим.ред.]. В данном случае надо смотреть файл **KBackgroundface.h**, содержащий искомую функцию. Открыв его в текстовом редакторе, вы увидите примерно следующее (взято из KDE 3.5.6):

```
/** Change the wallpaper.
* @param wallpaper The (local) path to the wallpaper.
* @param mode The tiling mode. */
```

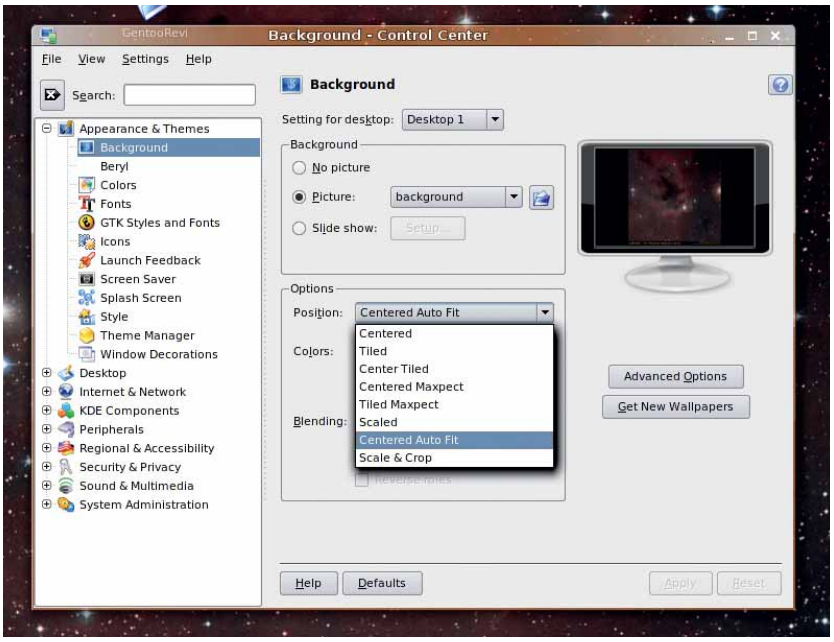
Это максимум, что удается извлечь, но кое-что про параметры DCOP понять можно: видно, что параметр mode определяет режим мозаики (tiling) при вставке изображения на рабочий стол. Именно он используется панелью *KControl* для изменения положения обоев в выпадающем меню Options. Использование DCOP часто требует подобной охоты за информацией о различных параметрах, но если вы знакомы с приложением, которым хотите управлять, то это не составит для вас большого труда.

Двойной щелчок на функции в *Kdcop* открывает небольшое окно, которое попросит вас ввести каждый из параметров в скобках. Здесь стоит сперва протестировать DCOP и удостовериться, что он делает именно то, чего вы ожидали. В setWallpaper, например, есть ошибка: функция не позволит вам установить последний из режимов (названный в *KControl* Масштабировать и Обрезать). Предположительно, это из-за того, что возможность добавили в приложение, не связав с разделом кода, взаимодействующего с DCOP. Подобные ошибки делают работу с DCOP немного непредсказуемой.

Используя *Kdcop*, вы можете дважды щелкнуть на setWallpaper, ввести путь к изображению и число 1 для режима и посмотреть как изменится рабочий стол. Удостоверившись, что это работает, можете попробовать версию из командной строки, составляя команду пути к функции. Вот что вам надо набрать, чтобы изменить фон рабочего стола из командной строки:

```
dcop kdesktop KBackgroundIface setWallpaper image.jpg 1
```

Вам также следует предоставить полный путь к изображению: функция setWallpaper принимает расположение файла чисто как строку символов, и не воспринимает относительные пути.



Теперь у вас должно быть хорошее понимание того, как работает DCOP и что он умеет. А значит, теперь мы готовы сделать что-то более полезное, чем переключение фона рабочего стола.

› Иногда без GUI невозможно понять, что делает очередная DCOP-функция: например, поле Position из KControl в DCOP называется Scale mode.

Менеджер виртуального рабочего стола

Одна из возможностей, которых мне всегда не хватало в KDE – группировка приложений на одном виртуальном столе в соответствии с задачей, которую они выполняют. Я бы выделил виртуальный рабочий стол каждой задаче: например, один для *KMail*, другой – для *Konqueror*, а третий – для терминала. DCOP позволяет набросать скрипт для вашего собственного решения; это простая альтернатива исследованию исходного кода и ручного добавления новых возможностей – что-то типа быстрого прототипирования KDE-приложений. Надеюсь, некая форма инструмента управления виртуальным рабочим столом войдет в состав KDE 4, но покамест вы можете построить свое решение с помощью DCOP.

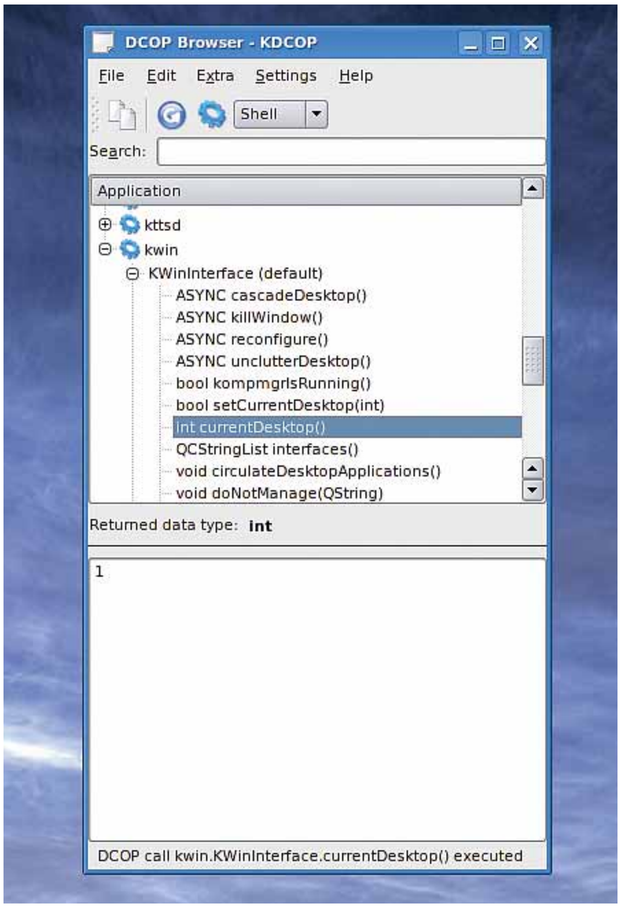
К несчастью, решение отнюдь не лежит на поверхности – одной DCOP-командой для переключения рабочего стола запущенного приложения тут не обойтись: команды, меняющей виртуальный рабочий стол приложения, в DCOP попросту нет. Об этом знают разработчики KDE, и хорошо бы такая команда появилась в KDE 4. Будем делать революцию с теми командами, какие у нас есть! А именно, спрячем приложение, которое мы хотим переместить на другой рабочий стол, и перейдем на требуемый стол, не восстанавливая приложения, а восстановим его до исходного размера на новом рабочем столе. Согласен, решение жутко неэффективное, но для того, кто будет смотреть на ваш экран, это будет круто, так как ваши окна будут перетасовываться автоматически.

DCOP-команды для свертывания и восстановления окна можно найти в списке функций верхнего уровня под заголовком имя_приложения-mainwindow#1. Рассматриваемые функции – это void hide() и void restore(). Чтобы проверить, что они работают так, как вы ожидаете, можете либо дважды нажать на них в *Kdcop*, либо попробовать составить командный эквивалент, выглядящий так: »

Скорая помощь

Вы можете определить, какой рабочий стол использовать для запуска приложения с помощью инструмента *Kstart*. Например, kstart -desktop 3 konqueror запустит *Konqueror* на третьем рабочем столе.

› **Некоторые DCOP-функции возвращают значения, и *Kdcop* отобразит эти значения в нижней панели. Здесь он возвращает номер текущего рабочего стола.**




```
dcop konqueror-5887 konqueror-mainwindow#1 'hide()'
```

Параметр hide() надо поместить внутри апострофов; в противном случае, *Bash* будет неправильно интерпретировать скобки и команда не выполнится.

Но прежде чем писать скрипт виртуального рабочего стола, надо решить еще одну проблему. Когда есть более одного экземпляра приложения (в данном случае *Konqueror*), то к имени приложения следует добавлять идентификатор процесса. Это немного усложняет предсказание имени приложения, которое будет использовано в нашем скрипте. Нужен способ добавления идентификатора процесса к имени KDE-приложения. Одно из решений состоит в том, чтобы вывести сразу все процессы *Konqueror*, употребив специальный символ *, например:

```
$ dcop konqueror*
konqueror-5887
konqueror-5870
konqueror-724
```

а затем использовать каждую строку вывода в качестве ввода для цикла for в вашем скрипте *Bash*. Однако есть лучшее решение, использующее другой инструмент DCOP, который присутствует на каждом рабочем столе KDE: *dcopfind*. Этот инструмент создан для поиска и возврата DCOP-возможностей DCOP-клиента – например, поиска приложений, которые поддерживают инъекцию URL. Но он рассчитан на обработку многих клиентов, и его побочный эффект – возможность использовать символы подстановки для одновременного обращения ко всем экземплярам приложения. Например, вот как спрятать все экземпляры *Konqueror*:

```
dcopfind 'konqueror*' 'konqueror-mainwindow*' 'hide()'
```

Я нашел еще одну ошибку в KDE 3.5.6, когда каждый экземпляр не сворачивался, хотя и должен был. В данном случае мне пришлось выполнять эту команду дважды, чтобы свернуть все. Если у вас такая же проблема, попробуйте сделать то же самое.

Теперь, когда все экземпляры *Konqueror* спрятаны, надо переключить рабочий стол на нужный, чтобы потом восстановить каждое приложение на новом экране. Для этого есть DCOP-команда для *KWin*, оконного менеджера KDE по умолчанию. Единственная проблема состоит в том, что если у вас альтернативный оконный менеджер, вроде предоставляемых *Beryl* и *Compiz3D*, вы не сможете использовать DCOP для переключения рабочих столов. Тогда вам придется найти другие команды. Но это может скоро измениться, так как идет активная разработка включения *Beryl* и *Compiz* в *KWin*.

```
dcop kwin KWinInterface setCurrentDesktop 3
```

Эта команда переключит текущий рабочий стол на третий. Остался последний шаг – восстановить все приложения на новом столе:

```
dcopfind 'konqueror*' 'konqueror-mainwindow*' 'restore()'
```

В результате каждый ранее спрятанный экземпляр *Konqueror* появится на новом рабочем столе – то есть вы как будто перетащили все приложения *Konqueror* на один рабочий стол. Полностью скрипт будет выглядеть следующим образом:

```
#!/bin/bash
dcopfind 'konqueror*' 'konqueror-mainwindow*' 'hide()'
dcopfind 'konqueror*' 'konqueror-mainwindow*' 'hide()'
dcop kwin KWinInterface setCurrentDesktop 3
dcopfind 'konqueror*' 'konqueror-mainwindow*' 'restore()'
```

Эти четыре строчки – только начало. В настоящий момент скрипт можно рассматривать как рабочий прототип и доказательство концепции; правда, пригодиться он может только при довольно редких обстоятельствах. Очевидно, его можно улучшить не одним способом. Но первым делом мы должны добавить поддержку всех используемых приложений. В моем случае это будут *KMail*, *KDevelop* и *Kate*, и каждое из них можно пристроить, видоизменив команду dcopfind. Например,

```
dcopfind 'kmail*' 'kmail-mainwindow*' 'hide()
```

найдет и свернет главное приложение *KMail*. Если используется всего один экземпляр приложения, то, возможно, проще будет использовать обычную команду dcop – например, маловероятно, что *KMail* будет запущен более чем в одном экземпляре. Использование dcopfind тут не строго обязательно, но так лучше с точки зрения целостности.

Вы обнаружите, что у каждого KDE-приложения есть раздел mainwindow, который содержит тот же набор функций, а стало быть, также имеет функции сворачивания и восстановления, позволяющие переместить эти приложения на выбранный рабочий стол. Исключениями являются приложения, не имеющие главного окна, например, апплеты на панели задач или фоновые сервисы, но в любом случае вам незачем управлять их рабочим столом.

Если вы работаете с большим количеством приложений KDE, то вам, возможно, следует использовать внешний файл настройки для хранения списка приложений, которыми вы хотите управлять, и рабочих столов, на которых вы хотите их видеть. Затем вам потребуется изменить скрипт, чтобы загружать в цикле каждую строку файла и обрабатывать имя приложения так же, как мы делали выше (свертывая и восстанавливая каждое приложение).

Клевые идеи для DCOP

Если вы хотите внедрить в жизнь новые навыки DCOP, почему бы не попробовать следующие идеи?

» Одной из самых желанных возможностей в *Kopete* является функция «buddy pounce», чтобы *Kopete* автоматически посылал сообщение, когда собеседник выходит в сеть. Простой скрипт DCOP может выполнить эту задачу.

» Последние версии *Kopete* отключают автоматическую проверку орфографии, пока вы общаетесь с кем-либо в чате. DCOP может автоматически включить ее снова.

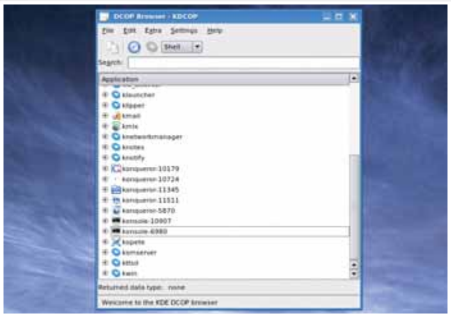
» В LXF79 была подробная статья о построении собственного Linux-киоска. DCOP является идеальным решением для блокирования рабочего стола вашего киоска, так как он даже может заставить переключиться Konqueror в полноэкранный режим киоска.

» *K3b* имеет один из самых лучших DCOP-интерфейсов, и там есть все виды работы с дисками, которые можно автоматизировать с помощью DCOP. Как насчет прожига ISO в режиме drag&drop?

Скорая помощь

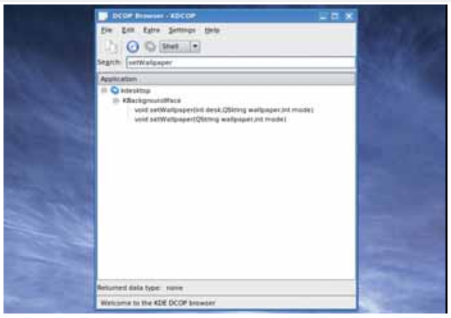
Всегда начинайте с *Kdcop*, чтобы удостовериться, что функции DCOP работают так, как вы ожидаете. После этого можете создавать собственные скрипты.

Шаг за шагом: Меняем обои на рабочем столе



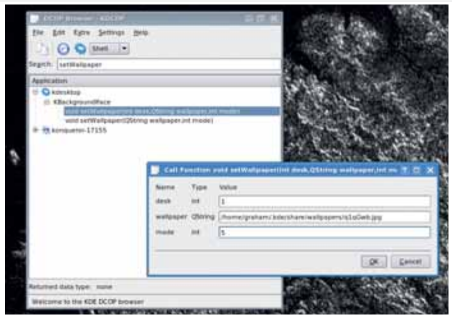
1 Запуск *Kdcop*

Kdcop необходимо запускать либо из командной строки, либо набрав KDCOP в панели Run Command. Основное окно покажет все запущенные KDE-приложения, а нажатие на каждом из них выведет список всех доступных функций.



2 Выбор вашей функции

Поле поиска отфильтрует список функций в реальном времени, и это лучший способ найти определенную функцию, не прибегая к ручному сканированию приложений. Функция, нужная для смены фона рабочего стола, называется setWallpaper.



3 Добавление свойств

Есть две функции setWallpaper. Вся разница между ними в том, что одна принимает два параметра, а другая три. Двойной щелчок по функции, и можно вводить требуемые данные – путь к изображению, рабочий стол, который вы хотите поменять, и режим масштабирования.

Последним штрихом будет возврат пользователя на исходный рабочий стол. Без этой модификации пользователь окажется на рабочем столе, где были восстановлены приложения последней группы, что вряд ли очень полезно. Такую функциональность можно реализовать через возвращаемые значения функций DCOP. Это единственный способ узнать, который рабочий стол вы использовали перед переключением на другой. Полученное значение может использоваться перед выходом из скрипта для возврата на прежние позиции.

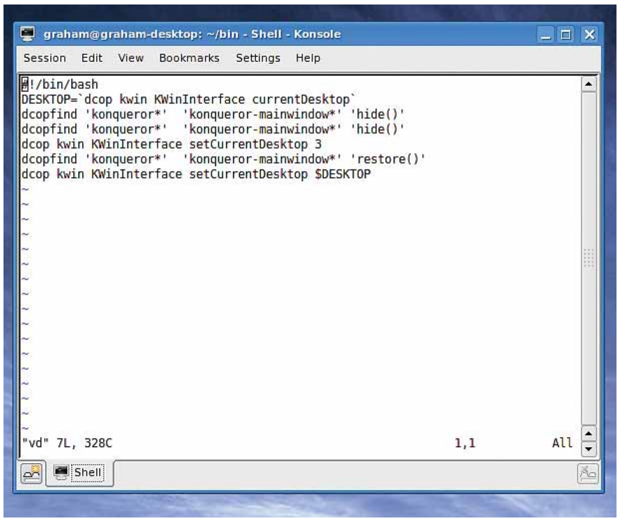
Вы могли заметить, что многие из функций DCOP начинаются со слова void. Обычно в этом месте указывается тип возвращаемого значения функции. Но когда программист использует слово 'void', это значит, что возвращаемое значение отсутствует, и функция просто отработает без какого-либо вывода. Функции, у которых вместо void проставлен тип, соответственно, возвращают значение данного типа. Например, значение функций, которые возвращают QStringList – несколько строк текста (программисты рассматривают текст как 'строку' символов), а значением функций, возвращающих int и float, будут соответственно целое число и число с плавающей точкой. Функция, определяющая номер используемого рабочего стола, возвращает целое значение. Она называется currentDesktop, и ее можно найти в иерархии под именем приложения KWinInterface в DCOP. Использование возвращаемых значений не составляет труда. Вот две строки, которые необходимы, чтобы сохранить значение начального рабочего стола в переменную DESKTOP, а потом переключиться назад по значению в DESKTOP:

```
DESKTOP=`dcop kwin KWinInterface currentDesktop`
dcop kwin KWinInterface setCurrentDesktop $DESKTOP
```

Первую строку надо поместить в начале скрипта, а вторую в конце. Теперь, запустив его, вы вернетесь на тот рабочий стол, откуда пришли, и как во всех хороших руководствах, конец – это только начало. В скрипте надо реализовать дополнительную проверку ошибок, чтобы убедиться, во-первых, что манипулируемые приложения действительно запущены, а во вторых, что столы действительно существуют. Обе этих функции добавить легко; сделайте это сразу после того, как заработает первоначальный скрипт.

Дерзайте!

У меня полным-полно идей по использованию DCOP, чтобы добавить функциональности KDE (см. «*Клевые идеи для DCOP*», напротив), и если у вас чешутся руки протестировать другой DCOP-проект, почему бы и не попробовать? Это также будет идеальным введением в программирование KDE, поскольку по мере написания скриптов DCOP вы изучите множество внутренних механизмов KDE. Если вы реализовали одну из наших идей или сделали собственную версию переключения рабочих столов, дайте нам знать. Мы будем счастливы включить ваши творения на DVD грядущих выпусков. LXF



› Финальный скрипт менеджера рабочего стола. Пока он поддерживает только Konqueror, но легко добавить и другие приложения.

» Через месяц Учитесь вместе с нами управлять пакетами через APT.

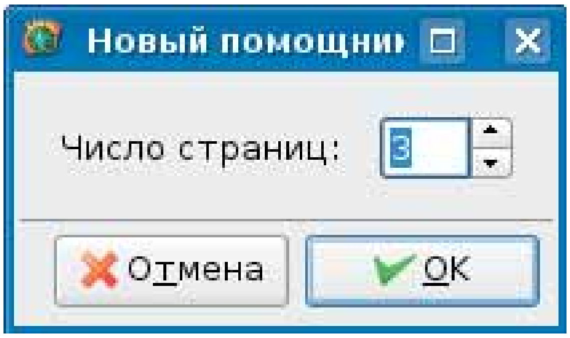
Заключительные

ЧАСТЬ 9 Пришла пора ставить точку — но напоследок **Андрей Боровский** изыскал возможность рассказать о двух блестящих компонентах GNOME и работе с *GTK* из C++.

Сегодня мы завершим беглый обзор основных библиотек GNOME, начатый в предыдущей статье, примером использования libgnomeui. Эта библиотека содержит несколько дополнительных визуальных компонентов, призванных, как сказано в документации, заставить интерфейс программы GNOME сверкать по-настоящему. В нашем примере мы рассмотрим два блистательных виджета из библиотеки libgnomeui – GnomeDruid и GnomeIconSelection.

Компонент GnomeDruid предназначен для создания мастеров (хотя в русском переводе интерфейса GNOME эти компоненты именуются «помощниками», я предпочитаю называть их именно так), сопровождающих пользователя через последовательные этапы некоего процесса конфигурации или выбора параметров. Пример использования GnomeDruid, который должен быть вам уже хорошо знаком – это мастер создания нового проекта в интегрированной среде *Anjuta*.

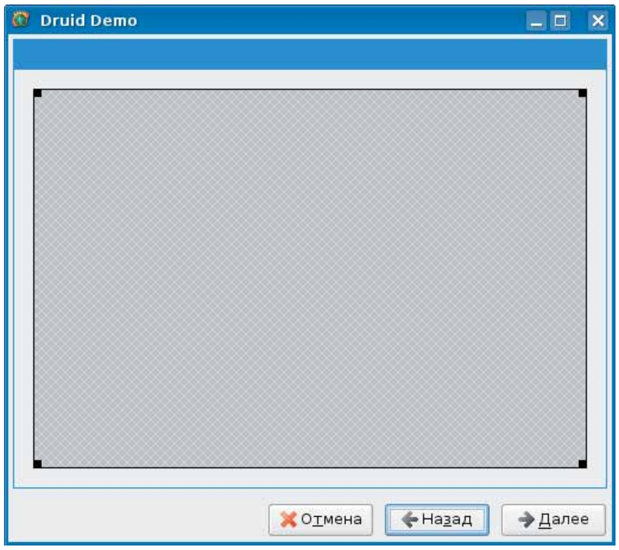
Второй из рассматриваемых компонентов, GnomeIconSelection, представляет собой окно, в которое может быть загружен набор пиктограмм. Этот компонент применяется в тех случаях, когда необходимо предоставить пользователю возможность выбрать пиктограмму из заранее заданного набора. Оба компонента относятся к категории контейнеров (GnomeDruid является потомком GtkContainer, а GnomeIconSelection – потомком GtkVBox).



› Рис. 1. Создание мастера (помощника) в среде Glade.

Создадим в среде *Anjuta* новый проект приложения GNOME с использованием *Glade* (программа, которую вы найдете на диске, называется druiddemo). В форму главного окна приложения помещаем компонент «Помощник» (то есть «мастер») со вкладки Gnome палитры компонентов *Glade*. При этом система спросит нас (**Рис. 1**), сколько страниц должен содержать создаваемый мастер (количество страниц можно будет изменить позднее, во время редактирования).

Рассчитывая количество страниц будущего мастера, следует учесть, что первая и последняя страницы обычно не содержат специальных элементов управления – они предназначены для вывода текста. Текст на первой странице обычно поясняет назначение мастера, а текст на последней странице поздравляет пользователя с успешным решением задачи и, возможно, объясняет, что следует делать дальше. Первая и последняя страницы содержат поля для вывода текста и заголовков, а также некоторые свойства, позволяющие задать дополнительные элементы оформления этих страниц. Все промежуточные страницы мастера, которые, собственно, и реализуют его функциональность, включают в себя поле для вывода заголовка страницы и контейнер GtkVBox, в котором вы можете размещать любые визуальные элементы (**Рис. 2**).



› Рис. 2. Страница мастера в режиме редактирования.

Переключение между страницами мастера во время редактирования выполняется так же, как и во время выполнения, то есть с помощью кнопок Назад и Далее. Наш демонстрационный мастер содержит всего три страницы, то есть в нем будет только одна страница с элементами управления. Поместив компонент-мастер в главное окно приложения, мы можем исследовать, какие объекты были сгенерированы для него *Glade*. Прежде всего, это объект druid1 класса GnomeDruid. Первая страница мастера представлена структурой druidpagestart1 типа GnomeDruidPageEdge. Свойство Заголовок этого объекта позволяет указать заголовок начальной страницы мастера, в свойстве Текст мы вводим текст.

Помимо этих двух свойств, класс GnomeDruidPageEdge обладает и другими, позволяющими указать цвет шрифта и фона для заголовка и текста, а также отобразить эмблему мастера в правом верхнем углу страницы и фоновый узор (watermarks) – в верхней части страницы. Страница, на которой мы будем размещать элементы управления, представлена структурой druidpagestandard1 типа GnomeDruidPageStandard. На ней расположен контейнер druid-vbox1 типа GtkVBox. У структуры GnomeDruidPageStandard тот же набор свойств, что и у структуры GnomeDruidPageEdge, за исключением свойства Текст (вместо него страница отображает содержимое контейнера).

» Месяц назад Мы разбирались с буфером обмена и виртуальной файловой системой GNOME.

штрихи

Последняя страница мастера, druidpagefinish1, является объектом типа GnomeDruidPageEdge, так же как и первая.

Помимо сигналов, унаследованных от своих предков, с объектами GnomeDruid связаны два специальных сигнала – help и cancel. Первый сигнал генерируется тогда, когда пользователь запрашивает справку; второй сигнал посылается приложению в результате щелчка по кнопке «Отмена», которая присутствует на каждой странице мастера. Мы назначаем обработчик сигнала cancel объекта druid1 (функция on_druid1_cancel()). Поскольку наша программа фактически состоит из одного мастера, будет вполне логично вызвать в нем функцию gtk_main_quit(). Объекты GnomeDruidPageEdge и GnomeDruidPageStandard, реализующие страницы мастера, обладают одинаковым набором дополнительных сигналов: back, next, cancel, finish и prepare.

Первые три сигнала посылаются приложению в ответ на щелчки по кнопкам Назад, Далее и Отмена соответственно. Сигнал prepare посылается приложению перед загрузкой соответствующей страницы, а сигнал finish – при щелчке по кнопке Применить, которая обычно расположена на последней странице мастера. Обратите внимание на то, что сигнал cancel есть как у объекта-мастера, так и у объекта-страницы. Зачем нужны два сигнала и как они взаимодействуют между собой? Если мы назначили обработчики обоим сигналам, то в ответ на щелчок кнопки Отмена на соответствующей странице будут, что вполне логично, вызваны оба обработчика. При этом сначала вызывается обработчик сигнала cancel объекта, реализующего страницу.

В то время как функция обработчика сигнал cancel объекта класса GnomeDruid возвращает значение void, функция обработчика cancel объекта страницы возвращает значение булевского типа. Если обработчик сигнала cancel объекта-страницы возвращает значение **TRUE**, обработчик сигнала cancel объекта класса GnomeDruid вызван не будет. Таким образом, назначая обработчик сигнала cancel объекта класса GnomeDruid и обработчик одноименного сигнала cancel для объекта страницы, мы можем запретить досрочное завершение работы мастера на соответствующей странице.

```
void on_druid1_cancel(GnomeDruid * gnomedruid, gpointer user_data)
{
  gtk_main_quit();
}

gboolean on_druidpagestandard1_cancel (GnomeDruidPage *
gnomedruidpage,
GtkWidget * widget, gpointer user_data)
{
 return TRUE;
}
```

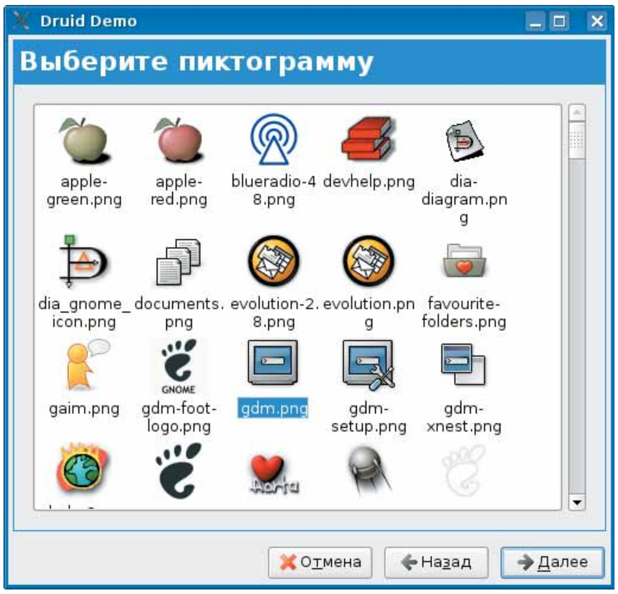
В нашем примере обработчик on_druidpagestandard1_cancel() запрещает завершение мастера на странице druidpagestandard1 (напоминаю характерную особенность интерфейса *GTK+*: когда обработчик сигнала должен запретить какие-либо дополнительные действия, он возвращает значение **TRUE**, в противном случае – значение **FALSE**). Впрочем, рассчитывать на этот метод запрета выхода из мастера особо не следует. Даже если обработчик сигнала cancel не позволит завершить работу мастера, пользователь все равно сможет сделать это, просто закрыв окно мастера с помощью кнопки «X», находящейся в его заголовке.

Сигнал finish посылается приложению тогда, когда пользователь щелкает кнопку Применить на странице GnomeDruidPageEdge. Как уже отмечалось, этот сигнал есть и у класса GnomeDruidPageStandard, однако при обычных условиях (без манипуляций с кнопками) промежуточная страница мастера этот сигнал не получит.

На странице druidpagestandard1 мы помещаем компонент GnomeIconSelection. Компонент GnomeIconSelection может быть очень красив, но в режиме редактирования мы увидим только чистое окно с полосой прокрутки. Во время выполнения программы мы должны указать компоненту источник пиктограмм для отображения и сделать отображаемые пиктограммы видимыми. Мы выполняем эти действия в обработчике сигнала prepare объекта druidpagestandard1:

```
void on_druidpagestandard1_prepare (GnomeDruidPage
*gnomedruidpage, GtkWidget * widget,
gpointer user_data)
{
 GnomeIconSelection * gis = lookup_widget(widget, "iconselection1");
 gnome_icon_selection_add_defaults (gis);
 gnome_icon_selection_show_icons(gis);
}
```

Функция gnome_icon_selection_add_defaults() добавляет в список пиктограмм iconselection1 набор пиктограмм GNOME, заданных по умолчанию. Функция gnome_icon_selection_add_directory() позволяет добавить набор пиктограмм из произвольно заданной директории. Мы можем добавить в список GnomeIconSelection несколько наборов пиктограмм. Чтобы сделать пиктограммы видимыми, вызовем функцию gnome_icon_selection_show_icons(). В результате окно компонента заполняется симпатичными пиктограммами GNOME (**Рис. 3**).



› Рис. 3. Компонент *GnomeIconSelection* с загруженными пиктограммами.

Компонент GnomeIconSelection позволяет не только разглядывать пиктограммы, но и выбирать их. В обработчике сигнала next объекта druidpagestandard1 мы проверяем, выбрал ли пользователь пиктограмму:

```
gboolean on_druidpagestandard1_next (GnomeDruidPage *
gnomedruidpage,
```

»

```
  GtkWidget * widget, gpointer user_data)
{
  GnomeIconSelection * gis = lookup_widget(widget, "iconselection1");
  char * icon = gnome_icon_selection_get_icon(gis, TRUE);
  if (icon != NULL) {
     strncpy(selected_icon, icon, 255);
     gnome_icon_selection_clear (gis, TRUE);
     return FALSE;
  }
  return TRUE;
}
```

Функция gnome_icon_selection_get_icon() возвращает имя файла выбранной пиктограммы или NULL, если ни одна пиктограмма не выбрана. Второй параметр функции указывает, должно ли имя файла пиктограммы содержать полный путь. Строка, возвращенная функцией gnome_icon_selection_get_icon(), не выделяется специально для нас, и мы не должны пытаться ее высвободить. Если пользователь выбрал пиктограмму, мы копируем ее имя в переменную selected_icon.

Как и обработчик сигнала cancel, обработчик сигнала next позволяет разрешить или запретить последующие действия с помощью возвращаемого значения. Пока пиктограмма не выбрана, мы не разрешаем пользователю переход на следующую страницу мастера.

Обратите внимание на функцию gnome_icon_selection_clear(). Эта функция очищает список пиктограмм. Если мы не будем очищать список пиктограмм каждый раз, когда пользователь покидает страницу с компонентом GnomeIconSelection, при каждом возвращении пользователя на эту страницу в компонент будет добавляться новая копия того же самого набора пиктограмм. По этой причине мы должны вызвать функцию gnome_icon_selection_clear() также и в обработчике сигнала back.

Когда пользователь переходит на последнюю страницу мастера, у него появляется возможность щелкнуть кнопку Применить. В обработчике соответствующего сигнала finish мы назначаем выбранную пользователем пиктограмму главному окну приложения:

```
void on_druidpagefinish1_finish(GnomeDruidPage * gnomedruidpage,
   GtkWidget * widget, gpointer user_data)
{
        GtkWindow * wnd = lookup_widget(widget, "window1");
        gtk_window_set_icon_from_file(wnd, selected_icon, NULL);
}
```

GTK+ плюс C++

Как отмечалось в самой первой статье этой серии, построение *GTK+* на основе C является определенным преимуществом, так как упрощает создание интерфейсов *GTK+* для других языков программирования. Однако вдумчивый читатель этих статей наверняка ловил себя на мысли, что объектно-ориентированная структура *GTK+* и GNOME API очень хорошо подходит для реализации на C++. Интерфейс C++ для *GTK+* и GNOME реализован в наборах библиотек *gtkmm* и *gnomemm*. Прежде чем приступать к разбору примеров *gtkmm*, убедитесь, что соответствующие пакеты установлены в вашей системе. Интерфейсы C++ для GNOME (*gnomemm*) не устанавливаются вместе с *gtkmm* по умолчанию. Их следует инсталлировать отдельно.

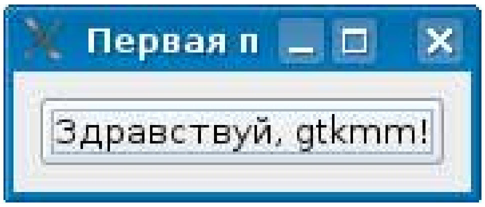
Рассмотрим простенькую программу, написанную с использованием *gtkmm* (файл **hellogtkmm.cpp** на диске):

```
#include <iostream>
#include <gtkmm.h>
class MainWindow : public Gtk::Window
{

        Gtk::Button m_button;
public:
        MainWindow(): m_button("Здравствуй, gtkmm!")
        {
                set_border_width(10);
                m_button.signal_clicked().connect(sigc::mem_
fun(*this,
                &MainWindow::on_button_clicked));
                add(m_button);
                m_button.show();
        }
        virtual ~MainWindow()
        {
        }
protected:
        virtual void on_button_clicked()
        {
                std::cout << "Button is pressed" << std::endl;
        }
};

int main(int argc, char *argv[])
{
        Gtk::Main prog(argc, argv);
        MainWindow window;
        window.set_title("Превая программа gtkmm");
        prog.run(window);
        return 0;
}
```

Эта программа делает то же, что и первая программа из первой статьи этой серии – создает окно с кнопкой (**Рис. 4**). Конечно, эта программа выглядит совсем не так эффектно, как предыдущая, написанная на «чистом» *GTK+*, но что уж тут поделать – изучение интерфейсов программирования развивается диалектично. Что дает нам использование интерфейса C++ для *GTK+*, написанного на C? Прежде всего – возможность более удобной группировки кода.



› Рис. 4. Программа *hellogtkmm*.

Главное окно нашей программы реализовано в классе MainWindow, который является потомком класса Gtk::Window (пространство имен Gtk содержит классы, связанные с визуальными элементами *GTK+*). Кнопка в нашей программе реализована с помощью объекта m_button класса Gtk::Button. В конструкторе класса MainWindow мы прежде всего создаем объект m_button. В параметре конструктора Gtk::Button передается текст кнопки. Метод set_border_width(), унаследованный классом MainWindow от Gtk::Window, делает то же, что сделала бы функция gtk_container_set_border_width() для объекта GtkWindow.

Здесь мы сталкиваемся с общим правилом «перевода» C-интерфейсов *GTK+* в C++ интерфейсы *gtkmm*. Допустим, в интерфейсе *GTK+* определена функция gtk_somewidget_do_something(), оперирующая данными структуры GtkSomeWidget. В *gtkmm* этой функции соответствует метод do_something() класса Gtk::SomeWidget. По правилам интерфейса *GTK+*, первым аргументом функции gtk_somewidget_do_something() должен быть указатель на экземпляр структуры GtkSomeWidget (или ее потомка), для которого вызывается функция. Метод Gtk::SomeWidget::do_something(), если он не статический, тоже получает в качестве первого аргумента указатель на объект класса Gtk::SomeWidget (или его класса-потомка), с той разницей, что в C++ передача указателя на объект, для которого вызван метод, выполняется неявно, через переменную this, так что в заголовке метода do_something() данный указатель можно опустить.

Если структура GtkOtherWidget является потомком GtkSomeWidget

в иерархии *GTK+*, (как например, структура GtkWindow происходит от GtkContainer), то соответствующий ей класс Gtk::OtherWidget является потомком Gtk::SomeWidget и наследует все доступные методы последнего. Таким образом, метод add() (с помощью которого мы добавляем кнопку в окно) класса Gtk::Window унаследован этим классом от класса Gtk::Container, в котором он соответствует функции gtk_container_add(), а метод show() объекта Gtk::Button унаследован от класса Gtk::Widget и соответствует функции gtk_widget_show().

Пространство имен Gtk становится доступно нам благодаря включению в текст программы файла **gtkmm.h**. Этот заголовочный файл включает в себя все заголовочные файлы *gtkmm*. С одной стороны, это хорошо, так как мы можем не опасаться, что забыли включить в текст программы объявление какого-либо нужного класса. С другой стороны, при использовании **gtkmm.h** препроцессору придется обработать около мегабайта заголовочных файлов, большая часть из которых конкретной программе не нужна. Поэтому в более сложных программах рекомендуется вместо одного универсального **gtkmm.h** включать в исходные тексты специализированные файлы (**gtkmm/main.h**, **gtkmm/button.h** и т.д.).

Пропустим, пока что, остальную часть конструктора и перейдем к функции main(). Здесь настоящее объектно-ориентированное программирование проявляет себя во всей красе. Любая программа gtkmm должна начинаться с инициализации объекта класса Gtk::Main (этот объект должен быть объявлен внутри функции main(), а не глобально). У класса Gtk::Main несколько конструкторов, мы выбираем самый простой из них, тот, которому передаются переменные argc и argv. Как вы, конечно, уже догадались, в программе может быть только один объект класса Gtk::Main. Система *gtkmm* сама следит за тем, чтобы в программе не появилось второго объекта Gtk::Main, так что если вы попытаетесь создать еще один объект этого класса, вы все равно получите ссылку на тот объект, который вы создали в начале [данная конструкция часто называется «синглетон», – прим. ред.].

Далее мы создаем объект класса MainWindow. Все, что касается внутренностей главного окна приложения (как в смысле визуальных элементов, так и в смысле данных), инкапсулировано в этом объекте, так что в главной функции нам достаточно передать ссылку (не указатель!) на объект класса MainWindow методу run() объекта Gtk::Main. В результате главное окно программы будет выведено на экран и начнет обработку адресованных ему сообщений графической системы. Закрытие главного окна приведет к выходу из метода run() и завершению программы.

Обратите внимание на то, что в отличие от *GTK+*, в *gtkmm* цикл обработки сообщений завершается вместе с закрытием главного окна. В приведенном выше примере метод run() был вызван как метод объекта prog для того, чтобы сделать код более понятным. На самом деле этот метод статический. Благодаря тому, что система *gtkmm* всегда «знает», как найти созданный в программе объект Gtk::Main, мы можем обращаться к нему, используя статические методы. Например, вместо

```
prog.run(window);
```

можно написать

```
Gtk::Main::run(window);
```

Вызов window.set_title(), как вы уже знаете, эквивалентен gtk_window_set_title() в *GTK+*.

Вот мы и научились «переводить» программные конструкции с языка *GTK+* на язык *gtkmm*. Это нетрудно, поскольку почти каждой функции *GTK+* соответствует метод какого-либо класса *gtkmm*. Но что делать, если нам все же понадобится вызвать функцию *GTK+* для объекта *gtkmm*? У класса Gtk::Widget и всех его потомков есть метод gobj(), который возвращает указатель на соответствующую структуру *GTK+*. Например, метод Gtk::Button::gobj() возвращает значение типа GtkButton*, которое можно передавать функциям *GTK+*.

Перегруженная функция wrap() из пространства имен Glib выполняет обратную операцию, то есть создает на основе структуры *GTK+* объект класса *gtkmm*:

```
GtkButton  * b = GTK_BUTTON(gtk_button_new_with_label ("fine button"));
Gtk::Button * button = Glib::wrap(b);
```

Вернемся теперь к конструктору MainWindow. Как вы, наверное, заметили, мы пропустили код, связанный с назначением обработчика сигнала clicked кнопки m_button. Прежде чем мы перейдем к описанию механизмов обработки сигналов *gtkmm*, позвольте небольшое лирическое отступление. Простые идеи обработки событий, заложенные в Delphi, Borland C++Builder и C#, почему-то не в чести у разработчиков открытых наборов графических компонентов. Эти парни не любят простых путей! Вспомните систему сигналов и слотов в *Qt*, требующую применения макросов и специального препроцессора...

Разработчики *gtkmm* шагнули еще дальше и задействовали для обработки событий «тяжелую артиллерию» C++ – функторы, адаптеры и шаблоны. Все управление обработкой сигналов в *gtkmm* выполняет библиотека *libsigc*. Если вы хотите досконально разобраться в том, как создаются, живут и умирают обработчики сигналов, изучайте документацию к этой библиотеке (которая, кстати, не слишком хороша). Нижеизложенное является лишь мягким введением в вопрос.

У каждого класса *gtkmm*, реализующего визуальный элемент *GTK+*, есть набор методов, соответствующих сигналам, которые может генерировать данный визуальный элемент. Каждый из этих методов возвращает прокси-объект, созданный на основе шаблона sigc::signal, его можно использовать для назначения обработчика сигнала.

Метод signal_clicked() объекта m_button возвращает прокси-объект для назначения обработчика сигнала clicked.

Метод connect() прокси-объекта осуществляет фактическое связывание обработчика и сигнала. Аргументом метода connect() может быть объект-слот или объект-функция (называемая также «функтор»), который мы создаем с помощью адаптера sigc::mem_fun(). Этот адаптер создает объект-функцию из метода класса (первым аргументом адаптера является указатель this, вторым – адрес метода). Если бы обработчик сигнала был обычной функцией, нам следовало бы воспользоваться адаптером sigc::ptr_fun(). У метода connect() есть еще один параметр булевского типа, который позволяет указать, должен ли обработчик вызываться после стандартного обработчика (по умолчанию этому параметру присвоено значение **TRUE**).

В результате всех этих манипуляций метод on_button_clicked() становится обработчиком сигнала clicked объекта m_button. Рассмотренный механизм позволяет назначать сигналу несколько обработчиков.

В общем случае метод connect возвращает объект, описываемый страшной конструкцией sigc::signal<void,int>::iterator. Этот объект легко преобразовать к типу sigc::connection. С помощью метода disconnect() объекта класса sigc::connection мы можем удалить назначенный сигналу обработчик:

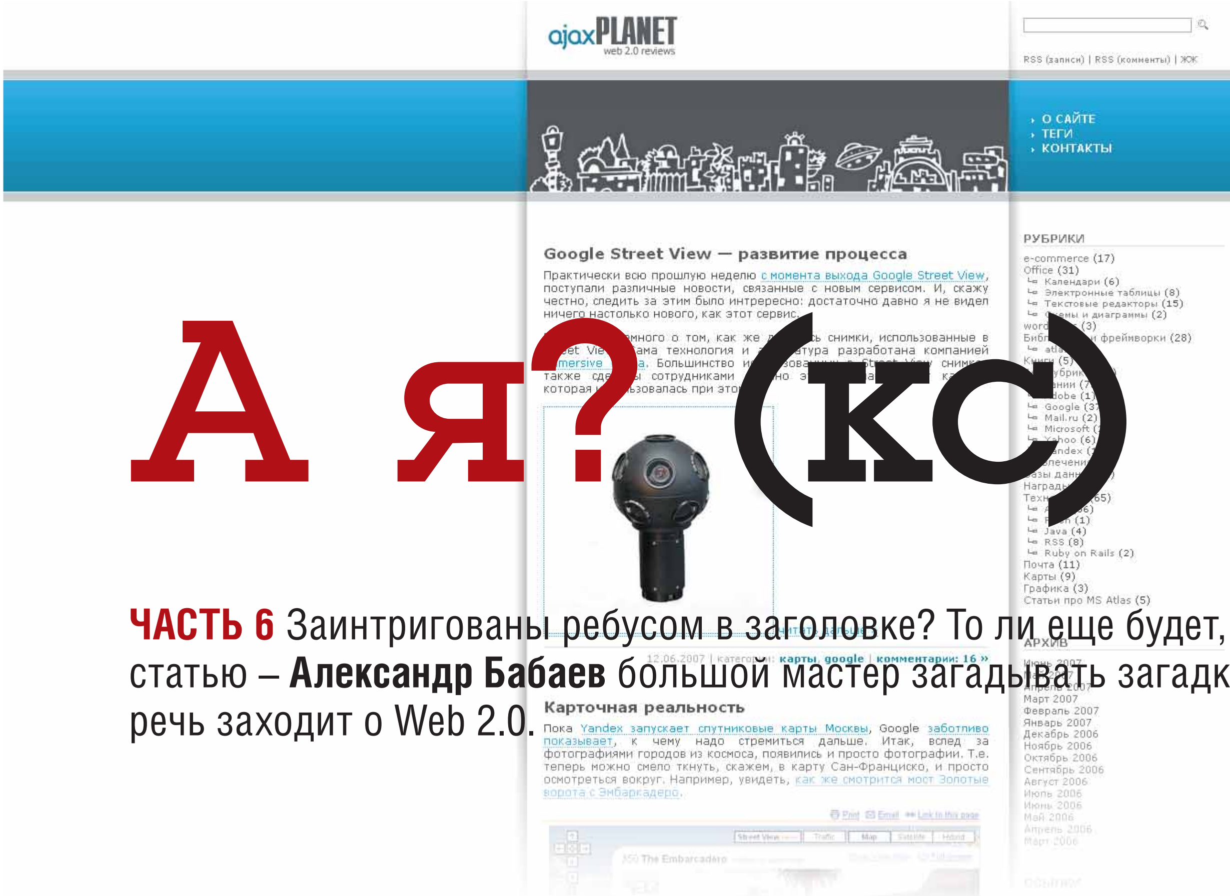
```
sigc::connection connection;
...
connection = m_button.signal_clicked().connect(sigc::mem_fun(*this,
                        &MainWindow::on_button_clicked));
...
connection.disconnect();
```

Если все это кажется вам слишком сложным, можете пойти другим путем. У класса Gtk::Button есть метод on_clicked(), объявленный в разделе protected. Вы можете перекрыть этот метод в потомке Gtk::Button (только не забудьте вызвать метод базового класса, иначе результаты могут оказаться несколько неожиданными). При таком подходе вам, конечно, придется создавать собственный класс для каждой кнопки вашего приложения [думается, что отчасти поэтому разработчики *GTK+* и *Qt* и решили не идти «путем Delphi», – прим. ред.].

Ну вот, пожалуй, и все, что можно было сказать о *gtkmm* в рамках одной статьи. Если вас интересует, как скомпилировать программу-пример командой в одну строку, то вот она, команда, использующая утилиту pkg-config:

```
g++ hellogtkmm.cpp -o hellogtkmm `pkg-config gtkmm-2.4 --cflags --libs`
```

На этом я завершаю увлекательное путешествие в мир *GTK+* и гномов и благодарю всех, кто не потерялся в пути. Встретимся вновь, изучая другие графические инструментарии.



А я? (кс)



ЧАСТЬ 6 Заинтригованы ребусом в заголовке? То ли еще будет, когда прочитаете статью – **Александр Бабаев** большой мастер загадывать загадки, особенно если речь заходит о Web 2.0.

Когда я начинал писать эту статью в первый раз, я думал, что все достаточно просто. Сел-написал-отдал. Но сразу после этого пришлось переделывать одно приложение, которое изменило понимание и сложности, и самого Аякса. Полностью.

В чем же сложность? В несоответствии. С одной стороны, Аякс позволяет приблизить логику работы интернет-приложений к обычной, «десктопной». С другой стороны – все браузеры таки разные. И это последнее «таки» местами убивает наповал. Вы думаете, что браузеров три? *IE*/*Firefox*/*Opera*? Хех. Для правильного приложения их нужно протестировать десяток. *IE 5.5/6/7*, *Firefox 1.5/2*, *Opera 8/9*, *Safari 1.3/2/* (а теперь и *3*). И у каждого свой нрав.

Ладно, что же такое Аякс?

AJAX – технология асинхронного доступа к серверу с использованием JavaScript и XML. Представим себе автомобиль. Чтобы заправить его – едем на заправку, заправляем, уезжаем. Чтобы поменять фару – едем, снимаем, ставим, уезжаем. Вот это – *AJAX*. А если бы Аякса не было, то для каждой такой операции нужно было бы отдавать машину в сервис, а из сервиса возвращалась бы другая, хоть и максимально точная копия вашей, с обновлениями (бак заправлен). Асинхронность – это вообще что-то вроде дозаправки в воздухе. То есть обновление данных без «отрыва от просмотра страницы». Давайте подробнее рассмотрим компоненты этой технологии:

A

Асинхронность реализуется разными способами. XMLHttpRequest, фреймы, *Flash*. Есть и другие (те же апплеты), но они распространены меньше. Второй способ – самый «простой» для человека, знакомого с HTML. Создается ма-а-аленький iframe (одна точка) и помещается куда-нибудь далеко, чтобы не было видно. Потом при необходимости подгрузить что-либо, запрос выполняется как обычно, но с целевым фреймом – тем самым, маленьким. Во фрейм загружается страничка, на которой обычно есть скрипт, который выполняется и делает свое дело (обновляет основной фрейм). Flash – это примерно то же самое, только вместо маленького скрытого фрейма используется маленькая «флэшка».

Впрочем, обычно оба эти способа используются только в случае, если не подходит «основной» – XMLHttpRequest. Это название объекта в JavaScript, который может выполнять асинхронные (и синхронные) HTTP-запросы. Например:

```
var xmlHttpRequestValue = new XMLHttpRequest();
xmlHttpRequestValue.open("POST", "/ajax.jsp?action=update", true);
xmlHttpRequestValue.onreadystatechange = processXmlHttpResponse;
xmlHttpRequestValue.send("Информация для пересылки серверу");
```

Все, запрос ушел. Сразу предупрежу, что это не рабочий код, только иллюстрация. Обратите внимание на третью строчку. Выделенное жирным – это имя функции, которая будет обрабатывать события, генерируемые во время передачи запроса, такие, как «начат прием данных», «закончен прием заголовка», «закончен прием данных». Если мы хотим что-либо предпринять по окончании приема данных, эта функция будет выглядеть как-то так:

```
function processXmlHttpResponse() {
  if (xmlHttpRequestValue.readyState == 4) {
     …сделать «что-либо»…
  }
}
```

Число «4» как раз обозначает, что обработка завершена. Функция processXmlHttpResponse() может вызываться много раз (даже точно не определено, сколько), но как только readyState станет равным 4, мы это «поймаем» и обработаем результат.

» **Месяц назад** Команды и фабрики – новый смысл старых слов в терминах ООП.

J

JavaScript – объектный язык. То есть создать класс, как в Java, в нем нельзя. А вот новый объект – запросто. И сформировать структуру объекта (прописать методы, поля) «на лету» тоже можно.

Чаще всего JavaScript используется как простой способ динамически изменять структуру документа. Например, чтобы поменять цвет параграфа, можно выполнить такой код:

```
document.getElementById("paragraph").style.color = "red";
```

Параграф после этого покраснеет. Чтобы понимать, как добраться до отдельных параграфов, нужно изучить DOM (Document Object Model).

X

XML, который попал в *AJAX* последней буквой, используется (как уже упоминалось) далеко не всегда, но часто. Благо, в JavaScript есть возможности по достаточно простому преобразованию чего угодно в XML, да и в других языках (не только в Java) библиотеки по работе с этим форматом достаточно хорошо развиты.

Помощники

Из-за того, что реализации XMLHttpRequest различаются (в паре крупных деталей и, что хуже, в паре десятков мелочей), самому писать кросс-браузерную обработку достаточно тяжело и долго. Поэтому давайте посмотрим на библиотеки, в состав которых входит «*AJAX*-подсистема» и которые сделаны либо специально для Java-северной составляющей, либо поддерживают ее.

» **DOJO** Огромная библиотека, которая много чего умеет. Если смотреть только на коммуникативные возможности, то тут тоже все в порядке: можно использовать XMLXttpRequest, фрейм, Flash. Первый вариант предлагается по умолчанию. Чуть ниже будет пример работы с DOJO, а сейчас просто отметим, что с учетом достаточно приличной документации, логичной организации и большого количества модулей (в том числе и для использования разных методов коммуникации), библиотеку можно назвать хорошей.

» **DWR – Direct Web Remoting**. Сделана специально для Java. Достоинство состоит в том, что DWR умеет «публиковать» классы Java в JavaScript. То есть вы пишете класс в Java, настраиваете DWR (при помощи XML-дескриптора), и после запуска сервлета (обработкой запросов от DWR занимается отдельный сервлет) в JavaScript «появляются» методы этого класса, которые можно вызывать, так же, как если бы это были обычные функции JavaScript. Способ удобен, но навязывает определенный способ коммуникации. Подходит, когда не нужно контролировать каждый байт передаваемых данных и если с JavaScript вы знакомы не очень хорошо (хотя все равно его нужно знать, так как кроме коммуникации есть много чего еще).

» **GWT – Google Web Toolkit**. В свое время эта библиотека наделала много шуму. Все дело в принципе ее работы. В общем и целом, если на DWR мы пишем на Java коммуникацию, то на GWT – все приложение. Затем GWT компилирует его в клиентскую часть (HTML + JavaScript) и серверную часть (Java). То есть клиент получается автоматически (не целиком, но близко к тому). GWT берет на себя все заботы и по взаимодействию клиента с сервером, и по созданию интерфейса (этот процесс несколько напоминает Swing), и многое другое. В принципе, достаточно хорошее решение, естественно со своими нюансами. Мы еще посмотрим на него поближе... к концу статьи.

Адресная книга с AJAX

Настало время потренироваться. Давайте возьмем нашу адресную книгу и сделаем так, чтобы показывалась страничка с табличкой (списком телефонов), а по нажатии на ссылки вместо перехода на новые странички менялся сам список: удалялись и добавлялись строки и так далее. Использовать будем DOJO. Идеология от этого не пострадает, но станет немного проще, короче и понятнее.

HTML/JavaScript

Вот строка HTML-таблицы, из которой видно, как будет выводиться список телефонов.

```
<tr id="1232323" >
   <td>1232323</td><td>Vasya Beanov</td><td><td>…</td><td><a
href="#" on click="remove('1232323'); return false;">Remove</a> / <a
href="#" on click="startEdit('1232323'); return false;">Edit</a></td>
 </tr>
```

Обратите внимание на id="…". Это своеобразные метки, которые позволят при необходимости найти нужную строку. Теперь рассмотрим JavaScript-код, который выполняется при нажатии на «кнопку» Edit (как самую сложную).

```
function startEdit(aPhone) {
   getCell(aPhone, 0).innerHTML = '<input type="text" value ="' +
getCell(aPhone, 0).innerHTML  + '" id ="phone_' + aPhone + '"/>';
   getCell(aPhone, 1).innerHTML = '<input type="text" value = "' +
getCell(aPhone, 1).innerHTML + '" id = "name_' + aPhone + '"/>';
   getCell(aPhone, 2).innerHTML = '<input type = "text" value ="' +
getCell(aPhone,  2).innerHTML + '" id ="comment_' + aPhone + '"/>';
   getCell(aPhone, 3).innerHTML = '<a href = "#" onclick = "submitEdit(\''
+ aPhone + '\'); return  false;">Save changes</a>';
}
```

Простым русским языком этом можно выразить так: ячейки с текстом заменяются на ячейки с полями ввода.

После редактирования и нажатия на кнопку Save changes данные отсылаются на сервер, и в ячейках снова прописывается текст.

```
function submitEdit(aPhone) {
   getCell(aPhone, 0).innerHTML = document.getElementById('phone_' +
aPhone).value;
   getCell(aPhone, 1).innerHTML = document.getElementById('name_' +
aPhone).value;
   getCell(aPhone, 2).innerHTML = document.getElementById('comment_'
+ aPhone).value;
   getCell(aPhone, 3).innerHTML = '<a href = "#" onclick = "startEdit(\'' +
aPhone + '\'); return    false;">Edit</a> <a href = "#" onclick = "remove(\''
+ aPhone + '\'); return false;">Remove</a>';

   sendAJAXRequest("action=edit" +
               "&old=" + aPhone +
               "&phone=" + getCell(aPhone, 0).innerHTML +
               "&name=" + getCell(aPhone, 1).innerHTML +
               "&comment=" + getCell(aPhone, 2).innerHTML);
}
```

Теперь посмотрим, как отсылаются и принимаются данные. Для этого у нас есть показательная функция loadTable(), которая загружает табличку с сервера. Остальные функции только отправляют данные.

```
function loadTable() {
   dojo.io.queueBind({
      url : "/ajax?action=loadTable",
      method : "get",
      load :  function (aType, aData, aEvent) {
         document.getElementById("table").innerHTML = aData;
      },
      preventCache : true
   });
}
```

Обратите внимание на две вещи. Первая – использование dojo.queueBind – это местная реализация *AJAX*. Нам не нужно беспокоиться, что будет использоваться «внутри» (хотя контролировать внутренности тоже можно). Вторая – параметр load. Это функция, которая вызывается после загрузки данных с сервера (сами данные передаются в параметре aData). sendAJAXRequest выглядит абсолютно так же, только без параметра load.

Кстати, в queueBind'е сами параметры обрамлены фигурными скобками. Это как раз и есть объектность JavaScript. Тут создается объект с

полями url, method, load и preventCache, и уже этот объект передается в качестве параметра функции.

Серверная часть

А теперь посмотрим, как это все обрабатывать на сервере. Логика его работы осталась той же: опять действия. Только вместо выдачи большого количества HTML-кода (через шаблоны или напрямую), в большинстве случаев не выдается ничего. Вот, например, код действия удаления:

```
String phone = aRequest.getParameter("phone");
_recordsBook.removeRecord(phone);
```

Только действие создания таблицы телефонов выдает код таблицы (но только ее, а не всю страничку). Остальные методы только получают данные и сохраняют их. Изменение таблицы происходит на клиенте при помощи JavaScript (как – можно посмотреть на код submitEdit выше или в полных исходных текстах на LXFDVD).

GWT

А что делать, если вы вообще ничего не понимаете в JavaScript? Не расстраиваться, скачать GWT (он немаленький, но и перспектива читать 1000 с лишним страниц талмуда под названием «JavaScript: The Definitive Guide» тоже не вдохновляет, ведь правда? К тому же архив можно найти и на нашем DVD), установить. А дальше?

Дальше процедура очень похожа на то, что позволяет делать, например, Ruby on Rails (или аналогичные каркасы – см. стр. XX):

- Вы создаете некий код. Клиентский, серверный, шаблон странички. При необходимости также можно применять свои CSS-стили.
- Затем создаете специальный XML-дескриптор. Это файл, который описывает, что нужно запускать, что будет сервером.
- Запускаете компилятор GWT. Он читает клиентский код, создает HTML-странички (используя ваши шаблоны) и JavaScript-код, который будет связываться с сервером и выполнять запросы, которые были описаны ранее.
- Запускаете сервер GWT-приложений.

При этом вы не пишете ни строчки на JavaScript. HTML и CSS – да, приходится, но опять же достаточно немного. Посмотрим подробнее?

Структура приложения GWT

Для начала стоит создать HTML-файл, который будет «заготовкой» для странички. Вот его код:

```
<html>
<head><meta name='gwt:module' content='phoneBook.PhoneBook'></head>
<body>
<script language="javascript" type="text/javascript" src="gwt.js"></script>
<table>
  <tr><td>Phone: </td><td id="phoneCell"></td>
     <td>Name: </td><td id="nameCell"></td>
     <td>Comment: </td><td id="commentCell"></td>
  </tr>
  <tr><td colspan="6" id="buttonCell"></td></tr>
</table>
<div id="main"></div>
</body>
</html>
```

Обратите внимание, что ни одной кнопки нет. И таблицы нет. Зато есть элементы, которые помечены id, в них-то и будут вставляться компоненты GWT, которые здесь называются виджетами.

Виджеты

Сами виджеты будут вставляться уже из «Java-кода». Почему в кавычках? Просто это код только пишется на Java, но потом компилируется в JavaScript и выполняется в браузере. Из-за этого есть множество ограничений. Например, нельзя использовать конструкции Java 5, такие как новые циклы foreach, аннотации, обобщенное программирование (generics) и другие. Нельзя использовать все классы подряд – только те, для которых в GWT есть «реализация» (для **java.lang.*** и **java.util.*** – есть, и это сильно упрощает дело).

Код вставки виджетов выглядит примерно следующим образом:

```
Button button = new Button("Add new record");
RootPanel.get("buttonCell").add(button);
```

Первая строчка создает кнопку, вторая вставляет ее в элемент с id «buttonCell».

Список виджетов достаточно велик, и можно также создавать свои собственные. Например, для вывода таблицы контактов можно использовать *FlexTable*.

```
FlexTable table = new FlexTable();
```

Этот виджет умеет динамически изменяться, например, вот этак:

```
aTable.setText(aRow, aColumn, "Текст в ячейку");
```

Этот код вставляет в ячейку таблицы текст. Можно также вставить туда и другой виджет, например, так:

```
aTable.setWidget(aRow, aColumn, new Button("Edit"));
```

Для работы нужны не только виджеты, но и обработчики событий. Это тоже реализовано достаточно просто: через слушателей (см. LXF92), как в Swing. Вот как, например, добавляется обработчик нажатия на кнопку:

```
editButton.addClickListener(new ClickListener() {
  public void onClick(Widget aWidget) {
    код обработчика...
  }
}
```

Есть, правда, и кардинальное отличие – в асинхронности. Для этого перейдем к сервисам.

Сервисы

Для обработки клиентских запросов пишутся так называемые сервисы. Чтобы собрать сервис, создается два интерфейса (phoneBook.client.PhoneBookService и phoneBook.client.PhoneBookServiceAsync) и реализация (phoneBook.server.PhoneBookServiceImpl). В сервис выносятся методы, которые сервер выполняет по запросу клиента (у нас это добавление, удаление, редактирование записей и выдача списка записей).

В «асинхронном» интерфейсе прописываются те же методы, что и в «обычном», но с одним дополнительным параметром AsyncCallback async. Это «Callback», обратный вызов. Когда делается асинхронный вызов, выполнение кода (работа с пользователем) продолжается. А когда работа асинхронного запроса завершается, вызывается этот обратный вызов, который сообщает клиенту: «Товарищ, задание партии выполнено, список пользователей доставлен».

Поэтому обработчики событий (например, кнопки удаления записи) выглядят примерно следующим образом (показан только метод onClick соответствующего обработчика):

```
// вызов сервиса, второй параметр – это и есть обратный вызов,
// то есть то, что вызывается после выполнения запроса
PhoneBookService.App.getInstance().
           removeRecord(_phone, new AsyncCallback() {
  public void onFailure(Throwable aThrowable) {
     // ничего тут не будем делать
  }

  public void onSuccess(Object o) {
     // удаляем строку из таблицы, соответствующую телефону
     for (int i = 0; i < _table.getRowCount(); i++) {
       if (_table.getText(i, 0).equals(_phone)) {
          _table.removeRow(i);
          break;
       }
     }
  }
});
```

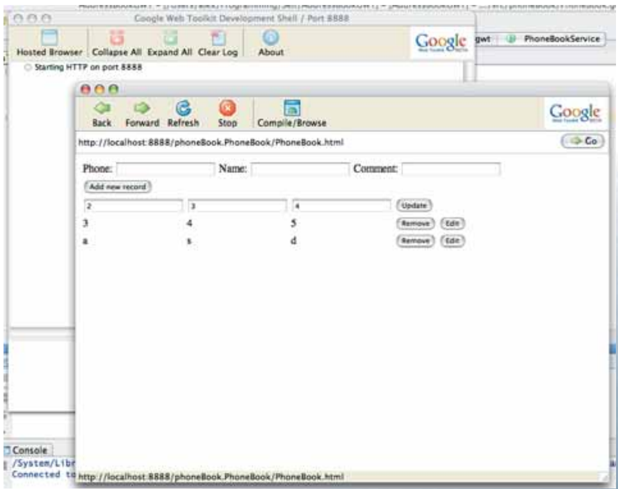
Компиляция и запуск

Ранее я говорил про дескриптор. Он очень прост и выглядит примерно так:

```
<module>
  <inherits name='com.google.gwt.user.User'/>
  <entry-point class='phoneBook.client.PhoneBook'/>
  <servlet path='/PhoneBookService' class='phoneBook.server.
PhoneBookServiceImpl'/>
</module>
```

Чтобы скомпилировать GWT-код, нужно запустить класс com.google.gwt.dev.GWTCompiler с параметром «полное имя класса, который нужно скомпилировать». Также можно указать параметр -out «каталог, куда складывать результат».

Запуск специальной оболочки для отладки осуществляется при помощи класса com.google.gwt.dev.GWTShell, которому в качестве параметра передается путь к HTML-файлу запуска. Более подробные строки компиляции и запуска лучше посмотреть в примерах, которые поставляются вместе с самим GWT. А как выглядит Google Web Toolkit Development Shell, можно узнать из рисунка – «вон он, змей, в окне маячит...»



› **Итог терзаний — все та же адресная книга и консоль на заднем плане.**

Ну, а все-таки, если делать самому?

Давайте поподробнее остановимся на некоторых подводных камнях технологии, связанных и с Java и с XMLHttpRequest'ом и с остальными ее компонентами.

Во-первых, Java (точнее, серверная сторона). Тут проблема одна, UTF-8. Почему-то некоторые браузеры не понимают UTF-8 в ответах. Safari, например. Есть достаточно простой обходной путь – выставлять не совсем правильный, но работающий тип (Content-Type) содержимого ответа (response), «text/plain; charset=utf-8».

Далее – клиентская часть. Тут подводных камней больше. Во-первых, количество соединений: на сайт браузер разрешает обычно не более двух. Поэтому «сделаем сразу 239 *AJAX*-запросов» не получится. Один-два в параллели, и все.

Во-вторых, выше упоминалось про callback, вызываемый, когда идет процесс загрузки ответа с сервера. Как и сколько раз он вызывается – четко не определено. То есть нельзя надеяться, что он вызовется с кодом события 4 ровно один раз.

В-третьих, на XMLHttpRequest не существует утвержденного стандарта или рекомендации, а есть только черновой вариант W3C. Поэтому реализации в разных браузерах различаются. Радует, правда, что не катастрофично.

Именно поэтому примеры приводятся с использованием сторонних библиотек. А если уж и нужно сделать «самому», то стоит воспользоваться опытом и посмотреть их исходные тексты. В этом-то и сила Open Source.

Вот и весь AJAX

В статье рассмотрен *AJAX* «по верхам». Тема безграничная, так как кроме технологии, использование *AJAX* меняет стиль и принципы работы web-приложений. Все то, как они работали до этого, должно быть переосмыслено на совершенно другом уровне. Дерзайте, и все получится. Клиенты будут довольны, а, следовательно, разработчики будут получать больше денег. LXF



Ссылки и интересные статьи

- **DOJO**: http://dojotoolkit.org,
- **DWR**: http://getahead.org/dwr,
- **GWT**: http://code.google.com/webtoolkit/,
- Статья про **GWT** от IBM developerWorks: http://www.ibm.com/developerworks/ru/library/j-AJAX4/index.html

» **Через месяц** Как обработать двадцать тысяч соединений в минуту? NIO!

«Клипы» и сессии

ЧАСТЬ 4 Нам часто приходится слышать, что открытое ПО хоть в чем-то да уступает своим проприетарным аналогам, и *Cinelerra* — не исключение. К счастью, **Петр Семилетов** знает, как превратить ее недостатки в достоинства....

Один из главных недостатков *Cinelerra*, наряду с больно уж простым менеджером импортированных материалов – это отсутствие аналога «последовательностей» (sequences) *Adobe Premiere Pro*. Последовательность – это, грубо говоря, единица монтажного стола, такой отдельный стол. Я монтирую одну сцену в одной «последовательности», другую – в другой, потом стыкую сцены вместе в третьей «последовательности». Это позволяет упорядочивать монтаж, делать его проще и удобнее. Сразу видно, где какая сцена. При монтировании можно оперировать уже этими цельными сценами (как обыкновенными фрагментами видео и аудио).

А в *Cinelerra* все надо монтировать на одном столе. Однако вы можете импортировать в текущую сессию другую сессию, сохраненную ранее. Подобное, кстати, сделано в *Sony Vegas*. Вы монтируете сцены А и Б как отдельные сессии *Cinelerra*. Затем в сессию В вы импортируете сцены-сессии А и Б и помещаете их на монтажный стол в сессии В, будто сессии А и Б – обыкновенные фрагменты видео или звука. Но с оговоркой! Когда вы помещаете целую сессию на монтажный стол, главное – чтобы хватило дорожек. Их должно быть достаточно – не меньше, чем в импортированной сессии. Потому что сессия импортируется не одной цельной дорожкой, а всем массивом содержащихся в ней дорожек, и материалы с этих дорожек должны поместиться на дорожках текущей сессии.

Стоит ли говорить, что параметры сессий должны совпадать? Я имею в виду частоту кадров, форматы и тому подобное.

Обратите внимание на способ импорта сессии в текущую сессию. Идете в File –> Load и в окне открытия файла, в нижнем списке (Insertion Strategy) выбираете пункт Create new resources only. Затем открываете нужную вам сессию, как любой другой файл. Сессия будет помещена в окно Resources, в раздел Clips.

А что такое эти Clips? Только ли сессия может быть клипом в терминологии *Cinelerra*? Нет. Клипом в *Cinelerra* называется поименованная совокупность видео и звуковых данных, расположенных на одной или нескольких дорожках.

Вы можете создать клип прямо на рабочем столе – выделяете область маркерами и нажимаете на панели инструментов кнопку To clip (или просто на клавишу i). Этот же инструмент есть в окнах Compositor и Viewer. Вызвав его, вы увидите окно, в котором клипу можно дать осмысленное имя и записать какие-нибудь примечания. Созданный клип будет доступен в окне Resources, в разделе Clips.

При этом важна одна особенность. Клип – не просто ссылка на какое-то место в монтаже. Это цельный объект. Например, где-то в конце монтажного стола вы отвели место под монтирование клипов. Вы монтируете клип, сохраняете его, а затем можете стереть смонтированные данные клипа с монтажного стола. Но при этом клип – как объект, как совокупность смонтированных материалов – останется по-прежнему доступен в Clips.

Клипы и сессии: Отличия

Когда вы помещаете сессию на монтажный стол, вы просто помещаете в монтажное пространство набор материалов, которые можно монтировать независимо друг от друга. То есть, вам доступны отдельно все составляющие импортированной сессии. Например, у вас в ней две дорожки – река и луна в небе. Поместив сессию на монтажный стол, вы можете передвинуть фрагмент видео с луной, или сократить фрагмент с речкой. Клип же помещается на монтажный стол как единое целое, хотя и занимает столько же дорожек, сколько и смонтированный в клипе материал. Грубо говоря, все материалы в клипе как бы намертво склеены между собой. Это проявляется при перемещении и изменении размера клипа.

Однако, вы можете разъединить составляющие клипа следующим образом: – активируйте кнопку Arm track для дорожки, которую вы хотите исключить из клипа, а затем сдвиньте клип в сторону. Та часть клипа, которая осталась на дорожке с выключенным Arm track, приобретет самостоятельность. Переключение Arm track также позволяет накладывать эффекты на отдельные дорожки клипа, не разрушая его целостности. Если вы отключили Arm track, потом наложили на дорожку эффект, а затем, не перемещая фрагмент клипа, снова включили Arm track, то фрагмент из клипа не исключается.

Клипы и сессии: Недостатки

Как же без критики? В *Adobe Premiere Pro*, если вы используете последовательность в качестве монтажного элемента, он занимает одну видео- и одну звуковую дорожку, то есть выступает как обычный фрагмент видео со звуком. И вы можете работать с последовательностью точно так же, как с любым другим фрагментом – например, накладывать эффекты на всю последовательность целиком. Да и перемещать последовательность так весьма удобно. В *Cinelerra* мы вынуждены перемещать многодорожечные клипы. А импортированная сессия вообще не доступна как цельный объект.

Допустим, я хочу поменять сцены местами. В *Adobe Premiere Pro* я беру одну «последовательность», тяну ее мышкой и ставлю следом за другой. В *Cinelerra* мне приходится выделять, вырезать и ставить в другое место весь массив смонтированных в сцене материалов. Кроме того, если я решил что-то подправить в сессиях А и Б, то в сессии В,

Месяц назад Мы узнали, как сделать свой собственный мультфильм.

куда я импортировал А и Б, изменения в этих двух сессиях никак не отразятся. Ведь импортированная сессия доступна как клип только в текущем сеансе!

Пример. В сессию Б мы импортировали сессию А. Сессия А становится доступна в списке Clips. Кроме того, в списке Media появляются все материалы из сессии А. Если вы сохраните сессию Б, закроете ее и откроете снова, то обнаружите, что в Clips уже нет никакой импортированной сессии А. Однако в Media остались все материалы оттуда; они же присутствуют и на монтажном столе.

Управление скоростью

Порой бывает необходимо изменить скорость видеофрагмента – ускорить его или замедлить. Если в *Adobe Premiere Pro* для этого предусмотрен особый инструмент, да и в свойства фрагмента можно зайти и задать там нужную скорость, то в *Cinelerra* эта задача решается другими двумя способами.

Способ первый, или обходной маневр. В папке Media окна Resources нажимаем правую кнопку мыши на фрагменте видео, скорость которого мы хотим изменить. В контекстном меню выбираем пункт Info. Появляется окно свойств файла, вот такое (**Рис. 1**).

Нас в нем интересует раздел Video, параметр Frame rate – частота кадров. Если выставить частоту кадров меньше настоящей (присущей видео изначально), то видео будет воспроизводиться замедленно. Если больше – то ускоренно. Обратите внимание, что можно не только выбирать предустановленные частоты кадров из списка, но и задавать

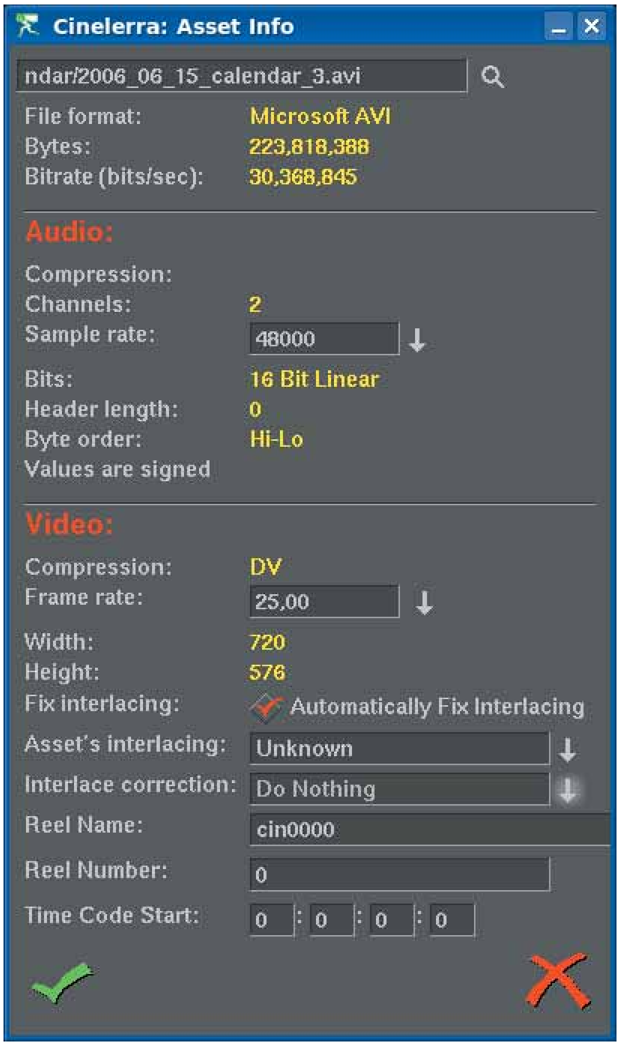


Рис. 1.

свои собственные частоты вручную. Например, если у вас есть видео с частотой кадров 25 кадров в секунду, а вы хотите ускорить его в десять раз, то поставьте новую частоту кадров – 250.

Важно помнить, что при изменении частоты кадров видимая протяженность видеофрагмента на монтажном столе не обновляется. То есть, вы увеличили частоту кадров, а фрагмент занимает на дорожке такое же расстояние, как и прежде. Надо изменять его размер вручную.

Другой способ управления скоростью видео – эффект ReframeRT. Чтобы ускорить видео, выбираем опцию Stretch и в поле ввода задаем число, большее единицы. Например, значение 2 означает «в два раза быстрее». А чтобы замедлить видео, опять выбираем Stretch, однако скорость задаем теперь в промежутке между 0 и 1. Например, 0.5 – половина изначальной скорости, то есть в два раза медленнее.

В окне эффекта ReframeRT есть еще режим Downsample. В отличие от ReframeRT, он не влияет на длину фрагмента, который обрабатывается эффектом. При Downsample значение в поле ввода показывает, сколь часто надо брать кадры из оригинала и выводить их с текущей скоростью проекта.

Пример. Есть видео с частотой кадров 30 кадров в секунду. Выбираем режим Downsample, ставим значение 0.5. Теперь частота кадров фрагмента виртуально падает до 15 кадров в секунду, т.е. уменьшается вдвое. Однако, настоящая частота кадров остается прежней, 30 кадров в секунду, просто из этих тридцати эффект каждую секунду берет и «показывает» только 15 кадров каждую секунду.

В принципе, ReframeRT – более гибкое средство изменения скорости, чем задание частоты кадров в свойствах файла, по двум причинам. Первая – эффект не влияет на настоящую частоту кадров видео, а меняет частоту лишь визуально (что достигается пропуском или дублированием кадров в зависимости от режима). И второе преимущество – вы можете обработать эффектом некую выделенную область, в то время как управление частотой кадров влияет на весь используемый видеофайл.

Обратите внимание, что все сказанное выше относится только к видеоданным. Даже если вы измените частоту кадров в свойствах файла, звук от этого не сожмется или не растянется во времени. И если в том же *Adobe Premiere Pro* изменение скорости фрагмента влияет и на связанный с фрагментом звук (хотя такое влияние можно отключить), то в *Cinelerra* звук придется подгонять под новую скорость отдельно, используя эффект вроде Time Stretch либо чего-то из модулей расширения LADSPA-плагинов.

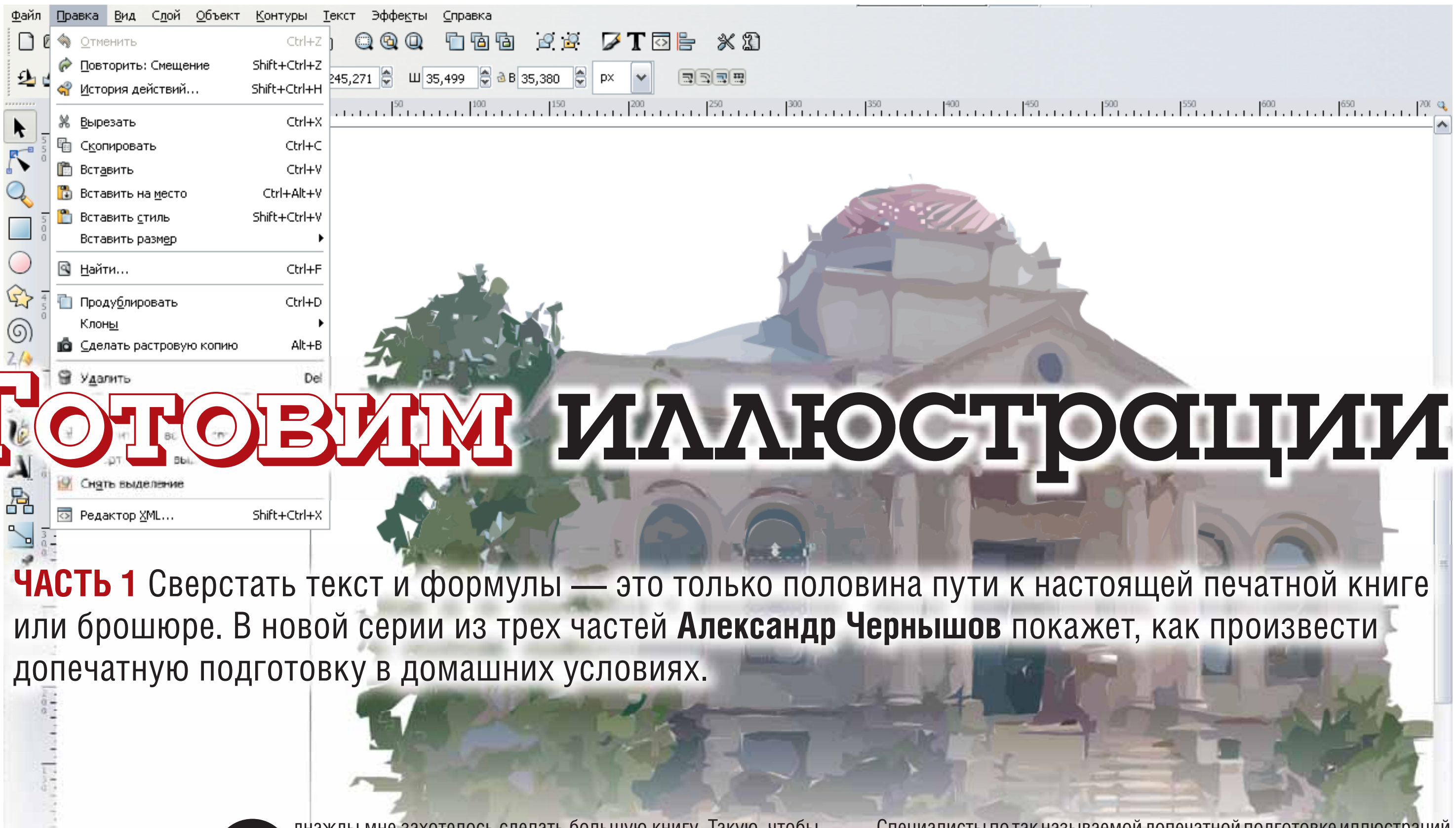
Можно также использовать эффект Reframe (без "RT" в названии), который доступен в меню Video –> Render Effect. С помощью этого средства вы можете обработать видео эффектом и «отрендерить» результат во внешний файл (либо последовательность статичных кадров, по одному на кадр), дабы потом вставить это обработанное эффектом видео вместо первоначального варианта. Разумеется, чтобы получить таким образом качественную картинку, не следует применять сжатие. А при DVD-разрешении несжатое видео весит ой как много!

На первый взгляд может показаться, что такой offline-рендеринг создан для слабых компьютеров с невероятно большими жесткими дисками (что само по себе – парадокс). На самом же деле его следует использовать для обработки фрагментов видео каким-нибудь ресурсоемким эффектом, который в реальном времени просчитывается с ощутимыми «тормозами», а вам нужно видеть результат обработки воочию, а не пытаться при низкой скорости отображения разобрать, что же там получается.

Сходное средство доступно не только для видео, но и для звука – ведь эффекты вроде реверберации могут потреблять довольно много вычислительных ресурсов. Для offline-обработки звука используйте Audio –> Render effect (появится список эффектов). Дорожка, к которой применяется эффект, должна быть помечена как armed, то есть надо включить на ней кнопку Arm track. LXF

Через месяц Титры: маленькая, но незаменимая деталь каждого серьезного фильма.

Новая серия! От вёрстки в TeX'е до профессиональной книги



Готовим иллюстрации

ЧАСТЬ 1 Сверстать текст и формулы — это только половина пути к настоящей печатной книге или брошюре. В новой серии из трех частей **Александр Чернышов** покажет, как произвести допечатную подготовку в домашних условиях.

Однажды мне захотелось сделать большую книгу. Такую, чтобы не стыдно было показать её кому угодно. Чтобы её было приятно взять в руки, листать и читать. Короче, чтобы выглядела она профессионально.

TeX позволил мне выполнить практически идеальную вёрстку текста (собственно, именно для вёрстки больших книг он и предназначен.). Но оказалось, что на этом путь к высокому качеству напечатанной книги только начинается.

В данном цикле статей я хочу поделиться с читателями проблемами, возникшими на этапе печати, и способами их решения, которые мне удалось найти. При этом я буду ориентироваться на систему *TeX*, но многое из написанного вполне применимо и к другим системам вёрстки, поскольку, когда дело доходит непосредственно до печати, особенности той или иной системы уходят на второй план.

Для подготовки и выполнения печати профессиональные полиграфисты используют дорогие проприетарные программы, работающие в Windows либо в MacOS. Между тем оказывается, что если только вы не собираетесь печатать суперспециализированное издание типа альбома картинной галереи, то получение профессионального результата вполне возможно с применением свободно распространяемых программ... И в Linux. Причём, вероятнее всего, все необходимые программы уже включены в ваш любимый дистрибутив.

Поговорим об иллюстрациях

Сколь аккуратно ни была бы выполнена вёрстка текста и математики, некачественно воспроизведённые иллюстрации, если они есть в тексте, способны испортить впечатление о любой блестящей работе. Помните, что *TeX* (как и другие системы) обработкой иллюстраций не занимается – он просто размещает в вёрстке заранее подготовленные материалы. И воспроизведением их он тоже не занимается – это дело сторонних программ просмотра и печати результатов вёрстки.

Мне доводилось видеть великолепно задуманные по идее и содержанию книги, содержащие уникальную информацию, напечатанные на высококачественной бумаге на офсете и имеющие дорогие переплёты. Но откровенные ошибки в подготовке иллюстраций (а точнее, игнорирование самой необходимости этой работы) портили впечатление так, что показывать книгу «в приличном обществе» становилось невозможно.

Специалисты по так называемой допечатной подготовке иллюстраций в профессиональной полиграфии недаром получают большие зарплаты. Но некоторые их «секреты» вполне доступны и непрофессионалам.

На сегодняшний день для подготовки вёрстки к качественной печати существует только один магистральный путь.

Смысл этого пути сводится к тому, что в мировой полиграфии уже давно в качестве стандарта де-факто используется язык PostScript (далее будет использоваться сокращение PS). А в составе практически всех распространённых сегодня дистрибутивов *TeX*'а имеется утилита (dvi-драйвер) *dvips*, которая позволяет преобразовать в формат PS весь DVI-документ, обеспечив тем самым возможность работы с документом на уровне международного стандарта.

Формат PS характеризуется двумя интересными свойствами:

- PS – это язык программирования. Документы PS содержат только текст и могут быть отредактированы в любом текстовым редакторе;
- формат прекрасно подходит для сохранения векторных иллюстраций.

Программировать непосредственно в PS вы вряд ли будете (хотя и это возможно) – для создания векторных иллюстраций существует большое число хороших векторных редакторов (мы поговорим о них чуть позже), а вот последнее свойство для нас весьма интересно. Векторные иллюстрации легко масштабировать без потери качества. Они не привязаны к разрешению печатающего устройства – следовательно, дают вам возможность отложить поиск типографии до окончания всей вёрстки.

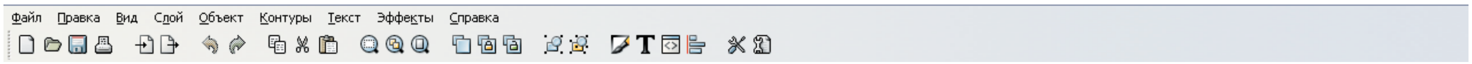
Иными словами, если иллюстрации для вёрстки представляют собой рисунки (чёткие линии и контуры, в частности, графики), имеет смысл приложить усилия к получению их в векторной форме. «Приложение усилий» будет зависеть от типа иллюстрации.

Рисунок можно нарисовать

Делается это в векторном графическом редакторе.

При этом, выбирая редактор, надо сразу проверить две вещи:

- возможность качественного сохранения результата в формате PS. Не все даже профессиональные редакторы, позволяющие сохранять в PS, делают это правильно. Например, знаменитый *AutoCAD*, по крайней мере в версиях для Windows, при сохранении в PS игнорирует толщины линий и создаёт ряд других проблем;



» возможность получить в PS-картинке русский текст (эта проблема долгие годы англоговорящим и англопишущим сообществом просто игнорировалась).

Проверка сводится к рисованию нескольких разных примитивов, вводу нескольких строк текста и сохранению рисунка в формате PS. При сохранении необходимо преобразовать текст в кривые (если редактор этого не позволяет, он вряд ли годится для работы).

После этого просмотрите картинку утилитой *gv*. Убедитесь, что в меню программы включено сглаживание (antialiasing), увеличьте масштаб до максимального и рассмотрите картинку по частям. Если видны все элементы рисунка (пусть даже с погрешностью на растр экрана), редактор использовать можно. Для полного счастья найдите лазерный принтер (не менее 600 dpi) и распечатайте эту картинку. Все вопросы должны исчезнуть.

Чтобы было меньше проблем при поиске принтера, конвертируйте картинку в PDF:

```
ps2pdf file.ps file.pdf
```

Теперь её можно распечатать из любого *Acrobat Reader*'а.

Лично я рекомендую для рисования *Inkscape*. Версия 0.43 уже вполне пригодна для серьёзной работы. Версию 0.44 на своём Debian 3.1 я запустить не могу – в его составе нет некоторых библиотек, без которых 0.44 жить уже не может.

Рисовать лучше в размере, близком к реальному размеру вставляемой в страницу картинки. Небольшие отклонения не страшны – картинка легко масштабируется.

Обязательно сохраните картинку в собственном формате *Inkscape* (SVG). И не забудьте записать этот SVG-файл в архив вместе с вёрсткой. Это даст возможность в будущем корректировать изображение или получать на её базе производные картинки. Сам формат PS не редактируется ни одним из свободно распространяемых инструментов.

Сохраняя картинку в формате PS (точнее, нас интересует EPS – encapsulated postscript, – специальный вариант PS для вставляемых картинок), не забудьте преобразовать текст в кривые. Это заметно увеличит объём файла PS-картинки, но обеспечит отображение в ней русских букв на любой системе (что нам и требуется!).

Если исходная картинка всё-таки потерялась, а откорректировать PS-картинку необходимо, можно попытаться воспользоваться утилитой *pstoedit*. К сожалению, несмотря на огромное количество форматов, которые может создать утилита, большинство из них по разным причинам для нас бесполезны.

Имеет смысл попытаться получить картинку лишь в формате SVG:

```
pstoedit -f plot-svg file.ps >file.svg
```

Предварительно проверьте, что в вашей сборке программы этот формат поддерживается:

```
pstoedit -f help
```

но не надейтесь на удобство редактирования – картинка по составляющим её элементам уже не та, что в первоначальном файле, да и все текстовые строки – это уже рисунки.

Так что не теряйте первоначальные рисунки!

Рисунок как копия экрана

Такой рисунок по определению является растровым. И работа с ним в общем случае требует особого подхода. Однако в некоторых случаях, затратив немного усилий, можно получить вполне приличный результат в векторной форме.

Прежде всего, запомните, что для полиграфии можно использовать только форматы, сжимающие без потерь. Стандартом здесь является TIFF, но можно использовать и PNG. Про JPEG забудьте и не вздумайте использовать его даже для промежуточного хранения ваших изображений!

Вообще говоря, существует примитивный, грубый и... неверный способ – использовать утилиту *convert* из состава *ImageMagick*:

```
convert file.png file.eps
```

считая, что файл с копией экрана имеет имя **file.png**. В результате будет получен настоящий PS, но содержащий немасштабируемую матрицу.

Устройство вывода не обманешь. Поэтому нам этот способ (пока) не подходит.

Во многих случаях копии с экрана содержат различные окна с меню, текстом и т.п., то есть с объектами, имеющими вполне чёткие границы на однородном фоне. Следовательно, имеет смысл попытаться их трассировать (векторизовать) и получить нормально масштабируемую картинку.

Строго говоря, проблема качественной трассировки произвольных изображений до сих пор нигде в мире не решена и надеяться на чудо здесь не стоит. Однако, копии с экранов различных окон и меню – тот случай, когда можно получить хороший результат практически бесплатно.

В арсенале современных дистрибутивов Linux имеется две программы для трассировки: *potrace* и *autotrace*. Если в вашем дистрибутиве их не оказалось, их легко найти в Интернете (**potrace.sourceforge.net** и **autotrace.sourceforge.net**, соответственно) и собрать из исходных текстов. Potrace трассирует только двухцветные (чёрно-белые) изображения, autotrace позволяет трассировать и цветные изображения.

Трюк здесь заключается в том, что изображение нужно правильно подготовить к трассировке.

Возьмём *GIMP* и с его помощью увеличим изображение вчетверо (пункт Масштабировать). Важно при этом сказать Интерполяция: Нет. Мы получим каждую точку, увеличенную по площади ровно в 16 раз. Возможно, то же самое можно сделать с помощью утилиты *convert* из *ImageMagick*, но я этого не пробовал.

Теперь запускаем:

```
autotrace file.png >file.eps
```

И проверяем, что получилось:

```
gv file.eps
```

Заметьте, что в большинстве случаев итоговый EPS-файл получится по размеру меньше, чем исходный с копией экрана. Вставить же его теперь можно на любую, большую или малую, площадь в странице без потери качества.

Рисунок-диаграмма или график

Его получают как результат расчёта в другой программе.

В этом случае вне конкуренции, конечно, *gnuplot*. Эта программа позволяет не только строить графики на экране, но и сохранять их в формате PS. Об этом уже довольно много написано (см., например, LXF71), поэтому от дальнейших комментариев воздержусь. Замечу только, что не надо забывать об общей проблеме комментирования графика русским текстом.

Если такая проблема возникла, то вот её решение.

Программу построения диаграммы запишите в отдельный файл. Комментируйте шкалы так, как вам надо, и не обращайте внимание на экранный предпросмотр. Когда всё будет отлажено, укажите выводить результат в формат SVG (!). Далее перекодируйте полученный SVG-файл в кодировку UTF8:

```
recode ..UTF8 <file1.svg >file2.svg
```

и загрузите **file2.svg** в *Inkscape*. Вы должны увидеть диаграмму с нормальными русскими комментариями. Теперь просто сохраните её в формате EPS.

Наконец, рисунок можно обвести по контуру

Этот способ особенно хорошо подходит для векторизации различных эмблем и товарных знаков. В распоряжение верстальщика эмблема чаще всего попадает в виде многократно скопированной, уменьшенной, увеличенной, получившей ненужные «ступеньки» от растра, потерявшей всякую форму «кляксы», не имеющей ничего общего с первоначальной идеей своего создателя.

Векторизация автоматическими средствами проблемы не снимет. Наоборот, она зафиксирует все эти уродливые растровые «лесенки», которых быть не должно. Поэтому единственный разумный способ – выполнить трассировку вручную, попытавшись приблизить результат к первоначальному замыслу. В качестве инструмента вполне подойдёт *Inkscape*.









» Для получения рисунков в PS можно, конечно, использовать и другие инструменты. Так, «гурманы» наверняка оценят *MetaPost* (см. LXF76-80).

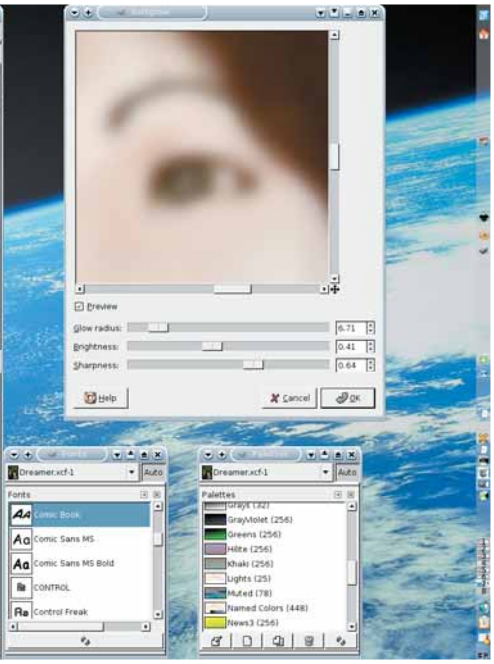
Вставка иллюстраций в документ

Ладно, иллюстрации в формате PS (точнее, EPS) получены. Как их вставить в документ?

Прежде всего надо проверить каждую иллюстрацию на границы (так называемый BoundingBox). Просмотрите каждую иллюстрацию через *gv*:

```
gv file.eps
```

Проверьте в меню *gv*, что стоит режим B.Box. Теперь оцените вид иллюстрации. Не должно быть больших белых полей (если только вы специально их не заказывали при получении EPS), линии рисунка не должны выскакивать за границы листа. Именно так, как вы видите картинку в *gv*, её будет воспринимать при вёрстке и *TeX*. Поэтому, если с полями что-то неправильно и «игра с параметрами сохранения EPS» в инструменте, которым получена иллюстрация, не даёт результата (что, к сожалению, бывает), то смело берите любой текстовый редактор, открывайте в нём наш EPS, ищите в первых его строках параметр




```
%BoundingBox xxx yyy xxx yyy
```

(xxx и yyy – числа в особых единицах, равных 1/72 дюйма, называемых «пунктами») и правьте эти числа, проверяя результат через *gv*.

Очень советую завести себе текстовый файлик и записывать сделанные исправления. Если придётся когда-нибудь повторить процедуру, вы по крайней мере, избежите проблем с подгонкой уже выполненной вёрстки.

После окончательной проверки иллюстрации можно вставлять в документ.

Для *plain TeX* существует простой макропакет *epsf*, который обеспечивает всю эту работу. Вот простой пример вставки иллюстрации **img.eps**:

```
\input epsf

% Какая-то часть документа

\midinsert
\epsfbox{img.eps}
\smallskip
Здесь подрисуночная подпись
\endinsert

% Продолжение документа

\bye
```

Изображение будет вставлено в своём размере «по умолчанию» и прижато влево. Текст подрисуночной подписи также будет прижат влево. Отцентрировать их командами \centerline{} проблемы не представляет.

Если надо подогнать размер иллюстрации под размер страницы, имеются команды \epsfxsize и \epsfysize. (И ещё несколько. Файл **epsf.tex** документирует сам себя. Загляните в него – узнаете много интересного.)

Имейте в виду, что хотя в некоторых системах некоторые версии *xdvi* позволяют увидеть вставленные иллюстрации сразу после получения DVI-файла, на самом деле полагаться на них не стоит. При некоторых «хитрых» вставках они сильно искажают результат, и вы будете долго искать ошибки там, где их нет.

Для просмотра результата после получения DVI-файла надо обязательно его перегнать в формат PS. Простейшая команда (только для предпросмотра!) выглядит так:

```
dvips -o file.ps file.dvi
```

Теперь **file.ps** можно просмотреть программой *gv*. При просмотре обязательно обращайте внимание на расположение иллюстраций на страницах и на их границы. Если что-то не так, это ошибки вставки. Качество самих изображений может вас не удовлетворить, но это следствие растеризации на мониторе. При печати на принтере качество должно быть максимальным (об этом мы ещё поговорим, когда доберёмся до печати).

Растровые иллюстрации

Это самый тяжёлый вопрос для полиграфии вообще. А уж для непрофессиональной – и подавно.

Прежде всего запомните – на допечатной подготовке растровых иллюстраций нельзя экономить время. Это огромный труд! Но результат обычно того стоит. Помните, что все случайные царапины и пыль, присутствующие на иллюстрациях, будут очень хорошо видны на итоговых оттисках!

В растровые иллюстрации превращается обычно всё то, что не удаётся качественно векторизовать, а также различные сканы и фотографии.

Для собственно вставки таких иллюстраций в документ существует единственный и уже упоминавшийся путь:

```
convert file.tiff file.eps
```

и далее – по уже описанной схеме. (Вместо *convert* подойдёт любая программа, позволяющая выполнить эту работу.) Поэтому наибольшее внимание должно быть уделено подготовке этого самого **file.tiff** (или **file.png**).

Процесс подготовки очень сильно зависит от того оборудования, на котором вы собираетесь печатать свою книгу. В каждой типографии печатное оборудование имеет свои уникальные особенности, которые надо знать, чтобы правильно подготовить растровые иллюстрации. Скажу больше, вам даже надо будет знать, на какой бумаге и какой краской вы будете печатать!

Поскольку для непрофессионалов знать и учитывать всё это довольно напряжённо, существует несколько более-менее универсальных приёмов сканирования и преобразования растровых иллюстраций, позволяющих получить удовлетворительный результат в большинстве типографий, не размениваясь на подробности об особенностях оборудования. Но суть этих процессов требует понимания существа дела. А его в одной короткой статье описать невозможно. Поэтому очень советую потратить некоторое время на изучение хотя бы следующих книг (или похожих):

» О'Квин, Д. *Допечатная подготовка. Руководство дизайнера*: Пер. с англ. – М.: Вильямс, 2003.

» Миронов, Д. *Ретушь на компьютере*. – СПб.: Питер, 2004.

Везде, где в этих книгах упоминается *Photoshop*, подставляйте *GIMP*. Однозначного соответствия друг другу возможностей этих программ нет. Однако, понимая суть процессов, вы без труда сможете выполнить (почти) все рекомендации в *GIMP*. И приготовьтесь к большим объёмам файлов иллюстраций. Маленькими объёмами высокого полиграфического качества в растре не добиться!

И ещё. Лично я пришёл к выводу, что имеет смысл самостоятельно готовить для вёрстки только чёрно-белые и полутоновые иллюстрации. Этот процесс вполне по силам любителям. И в большинстве случаев даже для очень хорошей книги чёрно-белых иллюстраций вполне достаточно. (Кроме всего прочего, чёрно-белая печать ещё и значительно дешевле!)

Для качественной подготовки цветных иллюстраций необходимо дорогостоящее специализированное оборудование. И за этот процесс лучше заплатить профессионалам. В конце концов, *TeX* сам по себе довольно плохо приспособлен для цветоделения, так что оставьте «кесарю кесарево». LXF

» **Через месяц** Офсет или ризограф: мы вплотную займемся печатью нашего произведения.



Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

1 Непонятки с удаленкой

Я в Linux уже давно, ставил еще Red Hat 5.1 на мой Amiga 4000, и изучил систему неплохо, но никак не могу запустить удаленный X-сеанс. Умею удаленно входить через SSH и работать в оболочке, но мне очень нужен доступ к моей удаленной машине именно через *X*.

Удаленный компьютер работает с *MythTV* на Kubuntu, а на том, с которого я хочу войти, установлен Gentoo. Доступ к рабочему столу нужен лишь для администрирования (не для просмотра *MythTV*), поэтому задача не кажется неразрешимой. Но я совершенно запутался: что считать сервером, а что – клиентом, что удаленной машиной, а что – хостом! У меня AMD64, некоторые его не хвалят.

Эндрю Уокер [Andrew Walker]

Эндрю, я тоже начинал с Linux на Amiga 4000 (с Red Hat 4.5) – помню, тогда все было намного сложнее, чем сейчас. Настроить удаленный X-доступ сравнительно легко, а в случае с *MythTV* еще и полезно, так как программа *mythtv-setup* может работать на удаленной машине, открывая при этом X-окно.

Применяя к *X* сетевую модель взаимоотношений «клиент-сервер», легко можно запутаться, если вы привыкли считать удаленную машину сервером, а свой компьютер – клиентом. *X*-сервер – это программа, создающая *X*-дисплей, и она работает на локальной машине. Клиентами же считаются программы, работающие на этом дисплее. Поэтому ваш Gentoo-компьютер – это сервер, а программы на машине с *MythTV* – клиенты. Запуск *X*-программы на удаленном сервере через SSH не вызывает затруднений и возможен при стандартных настройках SSH и на Gentoo, и на Kubuntu.

Войдите на Kubuntu-компьютер через SSH со своей Gentoo-машины с параметром **-Y**. вы получите возможность запускать там X-программы и заставлять



› Воспользуйтесь удаленным X-сеансом для запуска программы с одного компьютера на дисплее другого.

их открывать свои окна на рабочем столе Gentoo. Например, вот такие команды:

```
[user@gentoo]$ ssh -Y kubuntu
user@kubuntu's password:
[user@kubuntu]$ mythtv-setup
```

приведут к запуску на рабочем столе Gentoo программы *mythtv-setup*.

Иногда бывает, что после работы с X-программами сеанс SSH не заканчивается. Это значит, что какая-то программа запустила процессы, которые еще не закрыты: например, *KMail* открыла пару сокетов для подключения. С помощью *ps* в другом SSH-сеансе отловите и уничтожьте лишние процессы, тогда получите свое приглашение обратно.

Другие приложения, к которым вы можете обратиться – это инструменты общего доступа к рабочему столу, они или создают зеркальную копию, или открывают *X*-десктоп на удаленной машине. Для их работы необходимо, чтобы *X* был как минимум установлен на удаленном компьютере, а для зеркалирования он должен быть еще и запущен. Так как вы пользуетесь KDE, простейшим выбором для вас могут быть *krfb* и *krdc*. Первая из них – это сервер, работающий на удаленном компьютере и настраиваемый с помощью *KDE Control* Centre. Вторая запускается на локальном компьютере и отображает рабочий стол удаленного компьютера в окне. В Kubuntu обе программы установлены по умолчанию; в Gentoo-системе вам придется добавить *kde-base/krdc*.

VNC работает иначе, она предоставляет весь экран удаленному рабочему столу, отдельно от других экранных процессов локального компьютера. **НБ**

2 CentOS-конверсия

Согласно данным с сайта CentOS на www.centos.org, CentOS «намерен добиться 100%-ной двоичной совместимости» с продуктами «известной североамериканской Linux-корпорации». Тут я и призадумался. Можно ли перенастроить *Yum* системы Red Hat на хранилище CentOS? Во

Наши эксперты

» Мы найдем эксперта для ответа на любой вопрос, от инсталляции и установки модемов до администрирования сети – лишь черкните письмо, а об остальном мы позаботимся.



Нейл Ботвик
Поруководив небольшим Интернет-провайдером и поизготавливав сопроводительные диски для LXF, Нейл считает себя «Linux-мастером на все руки».



Майк Сондерс
Майк стоял у истоков Linux Answers в LXF. Его специализация – программирование, оконные менеджеры, init-скрипты и SNES.



Пол Хадсон
Пол – местный супер-программист, но справится также и с вашими проблемами с Сетью и базами данных.



Ник Вейч
Когда он не редактирует что-нибудь большим красным карандашом, Ник занимается графическим и трехмерным ПО для Linux. Готов ответить даже на самый простой вопрос!



Валентин Синицын
В редкие свободные минуты главный редактор нашего журнала обычно запускает mcedit, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите на м по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

время обновления CentOS-системы я заметил, что многие пакеты по-прежнему сохраняют имя Red Hat (например, **patch_for_что_то-RHEL-6.3.2**). Таким образом, можно было бы попытаться обновлять сервер после прекращения поддержки Red Hat. Я понимаю, что это не идеальный метод, но он хотя бы работает?

Джон К. Нолл [John K. Nall]

О Судя по сообщениям с форумов CentOS, такое вполне возможно, при условии, что осуществляется переход на эквивалентные версии, например с RHEL 5 на CentOS 5. У вас есть выбор: пользоваться хранилищами CentOS вместо Red Hat'овских, или полностью конвертировать инсталляцию Red Hat Enterprise Linux в CentOS.

Сначала убедитесь в том, что вы больше не зарегистрированы в Red Hat Network. Чтобы добавить хранилища CentOS, поместите в файл настройки Yum следующие данные:

```
[CentOS5 base]
name=CentOS-5-Base
mirrorlist=http://mirrorlist.centos.org/
?release=5&arch=$basearch&repo=os
gpgcheck=1
enabled=1
gpgkey=http://mirror.centos.org/centos/RPM-
GPG-KEY-CentOS-5

[CentOS5 updates]
name=CentOS-5-Updates
mirrorlist=http://mirrorlist.centos.org/
?release=5&arch=$basearch&repo=updates
gpgcheck=1
enabled=1
gpgkey=http://mirror.centos.org/centos/RPM-
GPG-KEY-CentOS-5

[CentOS5plus]
name=CentOS-5-Plus
mirrorlist=http://mirrorlist.centos.org/
?release=5&arch=$basearch&repo=centosplus
gpgcheck=1
enabled=1
gpgkey=http://mirror.centos.org/centos/RPM-
GPG-KEY-CentOS-5
```

Отключите RHEL-хранилища, заменив строку **enabled=1** на **enabled=0** для каждого из них. Эти действия включают *gpgcheck*, поэтому каждый пакет перед инсталляцией будет проверен на наличие ключей CentOS. Ключи можно установить так:

```
rpm --import http://isoredirect.centos.org/centos/5/os/
i386/RPM-GPG-KEY-CentOS-5
```

Если вы хотите полностью перейти на CentOS, понадобится установка двух небольших пакетов из хранилищ CentOS: **centos-release-5-0.0.el5.centos.2.x86_64.rpm** и **centos-release-notes-5.0.0-2.x86_64.rpm** или **centos-release-5-0.0.el5.centos.2.i386.rpm** и **centos-release-notes-5.0.0-2.i386.rpm**, в зависимости от Вашей системы (32 или 64-битной). Убедитесь в том, что удалены все ***-release-*** пакеты Red Hat.

Не исключено получение сообщений о конфликтах от Yum, вызванных тем, что в системе остается много пакетов RHEL-версий. Наилучшее решение в такой ситуации – конвертировать ее в чистую CentOS. Ведь у вас больше нет подписки на поддержку RHEL, поэтому бессмысленно беречь пакеты с маркой Red Hat, а полный переход на CentOS-систему позволит вам заручиться поддержкой CentOS-сообщества. **НБ**

3 Урезание OGG

В Можно ли уменьшить битрейт OGG-файлов? Я кодировал их на 458 кб/с, но они занимают слишком много места на диске.

hubris, из форумов LXF

О В спецификации Ogg Vorbis указано, что битрейт файла может быть снижен без перекодирования, но современное ПО пока не научилось осуществлять это, то есть каждый файл придется развернуть и закодировать снова. Это привело бы к некоторой потере качества при перекодировании с низких значений битрейта, но при таком высоком уровне, как у вас, потери будут минимальными. Если сохранились файлы-источники, то лучше всего было бы начать с нуля. В противном случае, единичный файл можно перекодировать вот такой строкой:

```
oggdec oldfile.ogg -o - | oggenc -q N -o newfile.
ogg -
```

N – значение качества. Задавая уровень кодирования не качеством, а значением битрейта, замените **-q** на **-b**. Конвертировать все файлы в директории можно так:

```
mkdir -p smalleroggs
for f in *.ogg
for FILE in *.ogg
do
 if oggdec "$FILE" -o - | oggenc -q N –o
"smalleroggs/$FILE" –
 then
  vorbiscomment -l "$FILE" | vorbiscomment –w
-c /dev/stdin "smalleroggs/$FILE"
 fi
done
```

Файлы будут перекодированы, а тэги старых файлов переместятся в новые. Если необходимо проникнуть в поддиректории, то понадобится команда **find** для поиска всех *.ogg-файлов. Приведенный вариант удаляет файлы-источники, поэтому соблюдайте осторожность.

```
find -name '*.ogg' | while read FILE
 NEWFILE=${FILE/.ogg/_new.ogg}
 if oggdec "$FILE" -o - | oggenc -q N -o
"$NEWFILE" -
 then
  vorbiscomment -l "$FILE" | vorbiscomment -w
-c /dev/stdin "$NEWFILE"
  mv -f "$NEWFILE" "$FILE"
 fi
done
```

ПХ

4 А теперь – PHP

В У меня проблемы с отображением PHP-файлов на моей Linux-машине (Mandriva 2007 Free). Вместо запуска, система упорно открывает их в *KWrite*. Я изучаю PHP, и, запуская HTML-файл, вызывающий PHP-процесс, совсем не хочу видеть код – мне надо, чтобы он, черт возьми, просто работал.

cynicalsurprise, из форумов LXF

»

Коротко про...

Файловые ссылки

Что такое жесткая (hard) и символическая (symbolic) ссылки?

В Linux широко используются файловые ссылки, в основном символические (гибкие). Файловые ссылки – это способ представить файл или директорию в двух или более местах или под разными именами. Можно просто скопировать файл в каждое из нужных мест, но у такого метода есть недостатки: расходуется дисковое пространство, а в случае обновления одного из файлов его копия не обновляется.

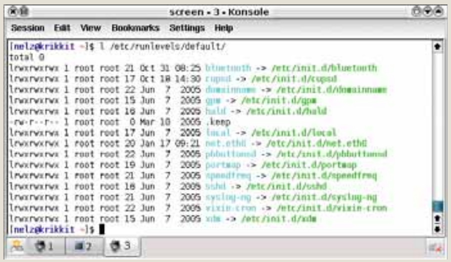
Файловые системы Linux обладают средством доступа к одному файлу или директории из разных мест – ссылками. Они бывают двух видов: жесткие и символические. С технической точки зрения, у каждого файла есть жесткая ссылка: это ссылка от имени файла к структуре данных на жестком диске, где хранится его содержимое. Говоря о жестких ссылках, мы имеем в виду дополнительные ссылки, когда один и тот же файл имеет больше одного имени. Возможности жестких ссылок ограничены: например, они могут ссылаться только на файлы, а не на директории, да и то в пределах единственной файловой системы.

Символические ссылки гораздо эффективнее, любая Linux-система буквально пронизана ими. Просмотреть такие ссылки можно, запустив **ls** с параметром **-l**, отображаются они и большинством файловых менеджеров. Классический пример – директория **/usr/src**, в которой **/usr/src/linux** – символическая ссылка на исходные тексты текущего ядра, например **linux-2.6.15**. Таким образом, у вас может быть инсталлировано несколько ядер, и каждая программа, которой необходим исходный код, будет искать свое в директории **/usr/src/linux**. Символические ссылки широко используются и в **/etc/rc.d**.

Создать символическую ссылку можно так:

```
ln -s /path/to/file /path/to/link
```

Один или оба пути могут быть относительными.



› **Много символических ссылок в /etc, файлы из которой должны быть доступны из разных директорий.**

» О PHP-файл через файловый менеджер вы не запустите. PHP – это скриптовый серверный язык, поэтому необходимо грузить PHP-страницу с сервера в web-браузер. Локально это простые текстовые файлы, вот файловый менеджер и делает то, что должен делать с такими файлами – просто открывает их в *KWrite*.

Это означает, что вам необходим собственный web-сервер – и не пугайтесь, не так страшен черт, как его малюют. Вызовите *Mandriva Control Center* (Центр управления Mandriva), перейдите в раздел инсталляции ПО, в поле поиска введите **mod_php** и выберите для инсталляции **apache-mod_php-5**. Будут установлены дополнительные пакеты, необходимые для обработки PHP-файлов. По окончании инсталляции перейдите в раздел System (Система) *Центра управления* и выберите пункт System Services (Системные службы). Убедитесь в том, что *httpd* (процесс *Apache*) настроен на запуск при загрузке, и если он еще не запущен, то запустите.

Укажите браузеру на http://localhost; должна появиться тестовая страница *Apache*, или, может быть, страничка 'It works!', подтверждающая наличие у вас работающего web-сервера. Теперь нужно поместить PHP-файлы в директорию **DocumentRoot**, где сервер ищет файлы для обработки. Mandriva по умолчанию использует **/var/www/html**, поэтому сохраните следующий файл как **/var/www/html/test.php**:

```
<?php
phpinfo();
?>
```

Загрузите http://localhost/test.php в браузер, и если вы увидите сведения о сервере и системе, на которой он работает, то *Apache* не только инсталлирован, но и настроен на обработку PHP-страниц, и вы можете смело продолжать изучение языка. Удачи!

При редактировании файлов в директории **DocumentRoot** могут возникнуть «правовые» проблемы Устранить их можно, добавив своего пользователя в группу *Apache* и открыв членам группы доступ для записи в эту директорию. Для этого наберите в терминале:

```
gpasswd -a yourusername apache
chgrp apache /var/www/html
chmod g+w /var/www/html
```

Чтобы изменения вступили в силу, может понадобиться выйти из своей учетной записи, а затем войти снова. **НБ**

Нейл не совсем прав. PHP-скрипты можно запускать из файлового менеджера – классическим примером является утилита *pptp-config*, которая не только работает, но и предоставляет графический интерфейс на базе *GTK*. Для этого необходимо просто добавить в начало скрипта строку **#!/путь/к/интерпретатору/php**, например **#!/usr/bin/php** и сделать его исполняемым (**chmod +x script.php**). Программы, ориентированные на web (а с ними, скорее всего, и имеет дело автор вопроса) работать не будут, да и само использование PHP в качестве языка общего назначения вызывает множество вопросов, но технически препятствий нет **ВС**

5 Вопрос о синхронизации

В Я установил Buffalo LS-250 LinkStation [сетевое хранилище] в свою домашнюю сеть, состоящую из Kubuntu Dapper и трех машин Windows XP. На ноутбуке Dapper нет никаких проблем с копированием файлов в обоих направлениях, да и настроить это было нетрудно.

Но! Мне хотелось бы синхронизировать ноутбук с LinkStation, а я не знаю, как это сделать. Я успешно настроил *Unison* для синхронизации ноутбука с одним из Windows-компьютеров, но не знаю, возможно ли это для LinkStation. Обратился было к *rsync*, но, похоже, для его работы надо установить дополнительное ПО и на ноутбук, и на LinkStation. Я бы обошелся командной строкой, написав скрипт для копирования в любом направлении только новых файлов, но ведь *rsync* именно для этого и предназначен.

А еще, на XP-машинах я открываю и редактирую файлы с LinkStation, а *Samba* позволяет открывать только копию на ноутбуке Dapper. Можно ли это исправить?

Ричард Мур [Richard Moore]

О Фактически, в Вашей домашней сети два Linux-компьютера, ведь LinkStation тоже работает на Linux. На http://linkstationwiki.net можно разыскать множество информации о различных моделях LinkStation, включая LinkStation Pro, как у вас. Наиболее интересным для вас может быть проект замены прошивки. FreeLink заменяет стандартный firmware-файл Debian-вариантом. Это более рискованно, чем OpenLink, но дает больше гибкости, хотя и за счет потери web-интерфейса. OpenLink базируется на firmware с добавкой некоторых программ, наиболее интересные из которых – SSH и *rsync*. Правда, ваши LS-LG – это новые модели, и на момент написания ответа OpenLink их не поддерживал, хотя ко времени публикации ситуация может измениться.

Если вы не хотите возиться со своим firmware, есть решение и попроще. Монтируя устройство с помощью *Samba*, вы можете использовать *rsync* без инсталляции чего бы то ни было на удаленную машину, как будто синхронизируются две локальные директории.

```
rsync -avx ~/myfiles/ /mnt/buffalo/myfiles/
```

Через SMB можно работать с файлами непосредственно на устройстве. Так как у вас KDE, то лучше сначала пойти путем KIO slave, открыв файл как **smb://name/path/to/file**. Попробуйте поискать файлы в *Konqueror* и открыть их в редакторе. Если не удалось, то это, скорее всего, следствие неверной настройки прав доступа и *Samba*. Запуская программы из оболочки, вы можете узнать более подробные сведения из сообщений об ошибках, возвращаемых системой. Например:

```
kwrite smb://name/path/to/file
```

НВ

Часто задаваемые вопросы...

Что такое руткит?

Не подцепите!

» Rootkit звучит безобидно, но я слышал, что это какая-то мерзость?

Верно: руткит – это комплект ПО, устанавливаемый кракером, чтобы скрыть свое присутствие на вашем компьютере. Например, модифицированные версии *ls* и *ps*, не отображающие процессы и файлы, принадлежащие крэкеру. Они обычно скрываются, как трояны, чтобы не вызывать подозрений, а их название обусловлено тем, что они инсталлируются и исполняются от имени суперпользователя (root). Зараженный компьютер описывается как 'rooted' (да не обидятся наши австралийские читатели: на их сленге это нехорошее слово).

» Какой ущерб может причинить руткит?

Сам по себе он почти безвреден. Его назначение – позволить хозяину-кракеру делать свое черное дело на вашем компьютере, не вызывая подозрений. Руткит изменяет программы для наблюдения за состоянием компьютера, чтобы те не подняли тревогу. В этом отношении руткиты отличаются от троянов.

» А это что такое?

«Троянский конь» (сокращенно «троян») – это вредоносная программа, маскирующаяся под нечто благовидное. Название происходит от уловки, использованной древними греками для проникновения в Трою. Отсюда и фраза «Бойтесь данайцев, дары приносящих» (а также добрых хакеров, предлагающих взломанное ПО). Adware (к счастью, пока не досаждающее Linux-пользователям) – характерный пример трояна, так как оно потихоньку «стучит хозяевам».

» Имеет ли это отношение к музыкальным CD?

Скандально известная программа DRM от Sony вполне укладывается в определение руткита, только для Windows. Программа была установлена такой известной компанией на такое большое число компьютеров, что ее вполне можно назвать величайшим (из известных) руткитом.

» Как догадаться о том, что компьютер заражен?

Существуют различные программы для проверки файловой системы на наличие признаков деятельности руткита. Они практикуют два метода: проверка кодовых последовательностей на соответствие находящимся в базе данных образцам руткитов или сравнение контрольных сумм MD5 различных файлов с данными, полученными при сканировании незараженной системы. Для успеха второго метода необходимо запускать такие программы сразу после инсталляции, пока система гарантированно «чиста». Вот две такие программы: *Rootkit Hunter* с www.rootkit.nl и *chkrootkit* с www.chkrootkit.org. Запускайте их регулярно – лучше с cron – когда компьютер подключен к Сети, и программы будут автоматически обновлять свои базы данных с образцами известных руткитов.

» А могут ли эти программы вылечить компьютер?

Нет. Как только руткит инсталлируется, компьютер теряет доверие. Руткит скрывает все следы деятельности кракера, чем бы он ни занимался, поэтому поражена может быть любая часть компьютера. Единственный выход – резервировать файлы данных и настройки и переустанавливать систему.

6 Слепой SUSE

Надеюсь получить простой ответ на простой аппаратный вопрос. Когда я пробую запустить SUSE 10.2 на моем новом 19-дюймовом плоскоэкранном мониторе, то получаю сообщение ‘not supported’ (не поддерживается). Как это исправить? На старом 14-дюймовом мониторе все работает прекрасно.

Пит Холлингс [Pete Hollings]

Аппаратная проблема, не связанная с проприетарными драйверами? Уже лучше! Значит, в центре экрана появляется это сообщение, и больше ничего? А это монитор сообщает, что сигнал компьютера выходит за пределы допустимого диапазона, так обычно бывает, когда компьютер пробует вывести изображение со слишком высоким разрешением или на слишком высокой частоте. Видимо, инсталлятор неверно распознает монитор и преувеличивает его возможности.

Так как вопрос касается лишь инсталлятора, простой ответ есть: предписать инсталлятору уменьшить значение разрешения. Для выбора разрешения нажмите F3 во время отображения загрузочного меню. Поднимайтесь по экрану (низкие разрешения отображаются в верхней части списка), пока не доберетесь до тех значений, с которыми монитор будет работать. В крайнем случае, можно инсталлировать в текстовом режиме. Это не столь приятно, к тому же требует некоторой сноровки, но полученная инсталляция будет абсолютно та же.

Проблема касается только инсталляции, вам нужно выбрать только такие параметры, которые позволят запустить графический рабочий стол. Уж он-то наверняка безошибочно определит ваш монитор. **МС**

7 На ощупь

Я хочу получить доступ к удаленному компьютеру, работающему без монитора, через удаленное подключение (*krdc*). Удаленная машина загружается без монитора, поэтому *X.org* постоянно откатывается на VGA (640x480). Можно ли заставить *X.org* работать на более высоком разрешении? Я не хочу довольствоваться удаленным *X*, мне нужен весь рабочий стол. На компьютере стоит Debian Etch, свой xorg.conf прилагаю.

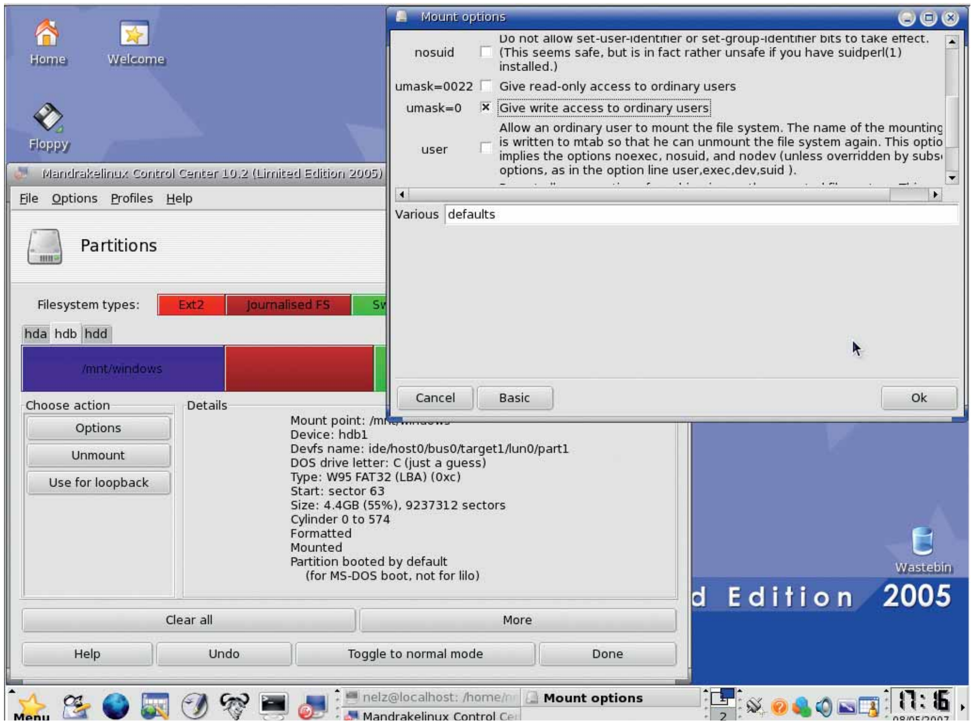
i.couchman, из форумов LXF

Такое поведение вызвано неверной настройкой Вашего *X.org*. Корень проблемы кроется вот в этом разделе **xorg.conf**:

```
Section "Monitor"
  Identifier  "BenQ T701"
  Option      "DPMS"
EndSection
```

Как видите, здесь не приводятся данные ни о возможностях, ни об ограничениях монитора. Такой подход постепенно становится стандартным и работает почти со всеми современными мониторами, поддерживающими EDID (Extended Display Identification Data, расширенную идентификацию данных дисплея). Во время настройки система опрашивает монитор и получает информацию, необходимую для правильной настройки дисплея. А если на монитор направить сигнал такой частоты или разрешения, которые превышают его возможности – хотя большинство современных мониторов имеет соответствующую защиту – то он может быть серьезно поврежден. Именно поэтому, не получив ответа на запрос EDID, *X.org* и возвращается к разрешению 640x480x8-бит.

Решение простое: сами добавьте информацию о частотах вертикальной и горизонтальной развертки, необходимые *X.org*, и он перестанет опрашивать несуществующий монитор. Добавить надо параметры HorizSync и VertRefresh к вышеприведенной секции **Monitor**. Если вы хоть раз подключали монитор к этому компьютеру, то сможете найти нужные значения в руководстве к монитору. Если же монитор вы никогда не подключали и не собираетесь этого делать, то можете взять любые разумные значения, да хоть из того же руководства, чтобы воспользоваться ими, когда это все же понадобится. Перезапустив X, вы увидите, что дисплей открылся на разрешении, указанном в **xorg.conf**, 1280x1024. **НБ**



› Центр управления Mandriva может изменить порядок монтирования Windows-раздела.

8 Верните Windows!

Недавно я установил Mandriva 2005 с вашего Специального выпуска [*Linux Format Special #1*], это уже во второй раз. В первый раз я мог обращаться к жестким дискам Windows, а сейчас – нет. Я словно перед запертой дверью. Как мне получить прежний доступ к этим дискам? Предыдущая инсталляция была на другом винчестере, у меня его больше нет.

Барри Симпсон [Barry Simpson]

Ответ зависит от двух обстоятельств: типа файловой системы, установленной на Windows-разделе, и Вашей интерпретации термина «запертая дверь». Если прежде вы имели неограниченный доступ к Windows-разделу, то, скорее всего, на нем была установлена система FAT32. В этом случае, если вы можете смонтировать раздел, но не можете записывать на него или входить в директории, то это просто проблема прав доступа.

Вызовите *Центр управления Mandriva*, перейдите в раздел Mount Points (Точки монтирования) и выберите Create, Delete And Resize Hard Disk Partitions (Создание, удаление и изменение размеров разделов). Выделите Windows-раздел, переключитесь в режим Expert и нажмите кнопку Options (Параметры). Поле в середине полученного окна будет содержать, скорее всего, ‘defaults’ (по умолчанию). Поставьте галочку в окошке с надписью Umask=0, затем OK и Done (Готово). Для применения настройки необходимо перемонтировать раздел. Можно было бы просто перезагрузиться, но ведь это Linux, а не Windows, поэтому откройте терминал и введите

```
su -c "mount /mnt/windows -o remount"
```

заменив **/mnt/windows** именем Вашего Windows-раздела. Введите пароль администратора, и вы получите всесторонний доступ к Windows-разделу. Все дело в параметре umask=0, добавленном к остальным параметрам монтирования. Файловая система FAT32 лишена собственных прав доступа, а упомянутый параметр позволяет системе рассматривать раздел как доступный для чтения и записи кому угодно.

Все усложняется, если на Windows-разделе установлена система NTFS. Доступ для чтения к этой файловой системе существует давно, но полный доступ освоен лишь в последнее время. Доступ для чтения осуществляется с помощью вышеприведенных шагов, только команду *remount* нужно заменить на

```
su
umount /mnt/windows
chmod 777 /mnt/windows
mount /mnt/windows
```

Теперь вы можете читать Windows-раздел – но не записывать на него. Хотя теоретически доступ для записи на Вашем дистрибутиве осуществить можно, но он весьма ограничен и не стоит затраченных усилий.

Mandriva 2005 сейчас считается устаревшим дистрибутивом, за прошедшее со времени его выхода время многое изменилось. Я рекомендовал бы обновиться до новейшей версии: Mandriva 2007 Spring можно инсталлировать с DVD нашего последнего номера. **НБ** »

Вопрос-победитель (английская версия)

О правах

В Мой Squid-бокс нормально работал до тех пор, пока скачок напряжения не уничтожил загрузочный сектор жесткого диска. Я переустановил систему (заодно обновил ее до Debian Etch).

Моя проблема в том, что при загрузке динамически создается папка **winbind_privileged**, причем с неверными правами доступа. Мне необходимы права root:proxy, чтобы прокси-сервер мог использовать авторизацию AD [Active Directory]. Как устранить эту проблему?

TwIsT, из форумов LXF

О Единственное надежное решение этой проблемы может показаться несколько безалаберным: позволить создание директории, а затем изменить групповую принадлежность. Запустите свой текстовый редактор как суперпользователь, откройте **/etc/rc.local** и перед завершающей строкой добавьте следующее:

```
if [[ -d /path/to/winbind_privileged ]]
then
  chgrp proxy /path/to/winbind_
privileged
fi
```

конечно, указав правильный путь к **winbind_privileged**. Этот скрипт запускается в самом конце процесса загрузки. Если вы хотите запускать его раньше, скажем, сразу после запуска Squid, понадобится отдельный скрипт. Поместите в **/etc/init.d/fixsquid** следующие строки:

```
#!/bin/sh
if [[ -d /path/to/winbind_privileged ]]
then
  chgrp proxy /path/to/winbind_
privileged
fi
```

Опять же, не забудьте подставить верный путь к **winbind_privileged**. Теперь сделайте скрипт исполняемым, а запуск сразу после старта Squid задайте следующими командами (под root'ом):

```
chmod +x /etc/init.d/fixsquid
ln -s ../init.d/fixsquid /etc/rc2.d/
S35fixsquid
```

Init-скрипты запускаются в алфавитно-цифровом порядке, а *Squid* запускается из **S35fixsquid**, следом запустится и наш скрипт (S означает startup; файлы, имена которых начинаются с K [Kill], запускаются при завершении работы, убивая процессы). **MC**

9 Бедствия Wi-Fi… опять!

В У меня проблема, в чем-то сходная со случаем Эндрю Вуда из LXF90 [Бедствия Wi-Fi, Ответы], только с адаптером D-Link DWL-G122 Rev C USB. Я пробовал *NdisWrapper* на другом D-Link DWL-G122, Rev B, и он работал на Windows-драйверах с сопроводительного диска. Но вот Rev C работать никак не хочет. Можно ли точно узнать, какой INF-файл следует использовать? Google сообщил, что на Rev C установлен чипсет Ralink RT73. Но как убедиться в этом локально, на Linux (Mepis)? Похож ли RT73 на чипсеты RT2x00?

Майкл Нг [Michael Ng]

О К сожалению, это очень распространенная проблема. Изготовители меняют внутренности устройств, не меняя их внешнего вида и названия. Пользователи Windows даже не догадываются об этом, пользуясь сопроводительными дисками с драйверами. вам необходимо придерживаться такой же линии с *NdisWrapper* – используйте INF-файл с сопроводительного диска (скорее всего, **rt73.inf**).

Идентифицировать устройство можно командой *lsusb*. Она вернет два шестнадцатеричных числа – идентификаторы изготовителя и продукта. Например, мой D-Link выдал

'Bus 001 Device 005: ID 2001:3700 D-Link Corp. [hex] DWL-122 802.11b'

где **2001** и **3700** – это идентификаторы изготовителя (VendorID) и продукта (ProductID) соответственно. Имея эти цифры, можно найти более подробные сведения на http://qbik.ch/usb/devices.

Чипсеты RT73 и RT25xx различны, хотя проект RT2x00 поддерживает и RT73, и RT61, только в другом варианте. На сайте RT2x00, http://rt2x00.serialmonkey.com, для RT73 доступен и отдельный пакет. Вероятно, его будет проще установить, вот и попробуйте. Загрузите архив **rt73-CVS**, распакуйте и следуйте инструкциям README-файла.

Если не сработает, попробуйте новый набор RT2x00, объединяющий драйверы для RT2400, RT2500, RT2700, RT61 и RT73 в один пакет. Так как он совершенно новый, загрузить его можно из репозитория Git, на странице загрузки есть ссылка на полные инструкции по установке.

Другой вариант – Linux-драйвер для RT73, доступный с сайта Ralink, текущий адрес www.ralink.com.tw/data/RT73_Linux_STA_Drv1.0.3.6.tar.gz. Архив содержит исчерпывающую инструкцию по установке. Понадобится компиляция драйвера из исходных текстов, то есть необходимо наличие инсталлированных исходных текстов ядра и GCC из стандартных пакетов Mepis. Необходимо также установить firmware-файл с www.ralinktech.com.tw/data/RT71W_Firmware_V1.8.zip.

По мере становления пакета драйверов RT2x00 ситуация должна выправляться. Я не поклонник *NdisWrapper*, но понимаю, почему им пользуются, ведь он зачастую «просто работает». **НБ**

10 Свобода общения

В У меня есть офисный маршрутизатор, работающий под управлением Linux и подключенный сразу к нескольким провайдерам. У каждого из них есть свои бонусы (бесплатный трафик на те или иные ресурсы). Маршрутизацию по хостам настроить несложно, но некоторые провайдеры предлагают бесплатный трафик по определенным портам: IRC (6667-6669), ICQ(443) и т.д. Перебирать все возможные хосты проблематично, да и некрасиво. Как в Linux осуществить маршрутизацию на основании порта пункта назначения?

Алексей Ярославцев

О Исходя из сказанного, я полагаю, что маршрутизация трафика между несколькими каналами у вас уже настроена и вопрос только в том, как осуществлять ее на основе порта назначения. Если это не так – прочтите раздел 4.2 в Linux Advanced Routing & Traffic Control HOWTO (**lartc.org/howto**): в нем вы найдете весьма подробное описание всех возможных тонкостей. Да и вообще, держите этот документ под рукой: когда провайдеров становится более одного, без LARTC не обойтись.

Сформулированная вами задача решается в два этапа. В первую очередь необходимо пометить интересующие вас пакеты – для этих целей в *iptables* предусмотрена цель MARK. Допустим, вас интересует трафик IRC. Тогда команда может выглядеть следующим образом:

```
iptables -t mangle -A PREROUTING --dport 6667:6669 -
p tcp -j MARK --set-mark 64
```

Данное правило предписывает установить метку '64' всем пакетам, порт назначения которых лежит в пределах **6667-6669/tcp**. Значение метки можно выбирать любым – главное, чтобы оно было уникальным в пределах системы.

Затем необходимо настроить политику маршрутизации. Для этого служит утилита *ip* из пакета *iproute2*. Мы должны проинструктировать ядро проверять значение метки на пришедшем пакете и, если оно равно 64, маршрутизировать его по особой таблице, единственная запись в которой предписывает использовать шлюз соответствующего провайдера. Итак, приступим.

» Придумайте название для таблицы маршрутизации – скажем, *IRCFriendly*, а также цифровой идентификатор (100). Их необходимо дописать в конец файла **/etc/iproute2/rt_tables**, при этом выдуманные вами значения, естественно, не должны совпадать с существующими:

```
echo 100 IRCFriendly >> /etc/iproute2/rt_tables
```

» Добавьте в таблицу *IRCFriendly* шлюз по умолчанию Поскольку стандартная команда route модифицирует только таблицу с говорящим названием *default*, воспользуйтесь утилитой *ip*:

```
ip route add default via 1.2.3.4 dev eth1 table
IRCFriendly
```

eth1 и **1.2.3.4** естественно, необходимо заменить на интерфейс, к которому подключен кабель провайдера и адрес его шлюза по умолчанию. Маршрут до узла **1.2.3.4** должен быть известен Linux – иначе команда выдаст ошибку.

» Заключительный шаг – заставить ядро маршрутизировать все пакеты, имеющие метку 64, согласно таблице *IRCFriendly*. Для этого нужно создать соответствующее правило:

```
ip rule add fwmark 64 table IRCFriendly
```

Вот, собственно, и все. При желании можно избежать правки файла **rt_tables**, напрямую указывая выбранный идентификатор таблицы во всех командах. **ВС**

Большой вопрос

На этом раздел заканчивается, но наши эксперты не расходятся по домам до тех пор, пока хотя бы один из них не дойдет до сути общей проблемы.

11 Обновить Debian без сети

В У нас есть много Debian-компьютеров без подключения к Интернет. Некоторые даже не в сети. Как лучше всего обновлять на них систему? В настоящее время мы копируем Deb-файлы на CD, затем вручную инсталлируем их на каждый компьютер, но я уверен, что есть способ получше. Мы подумывали о локальном Debian-зеркале, но такой метод съедает слишком много сетевых ресурсов, а для изолированных систем и вовсе бесполезен.

Ричард Кросс [Richard Cross]

О вам может помочь одна полезная штучка: пакет *APTonCD*. Программа создает хранилище на CD (или DVD), которым можно пользоваться для инсталляции или обновления изолированных машин. *APTonCD* также полезный инструмент клонирования и резервного копирования: создаваемые им диски содержат все пакеты, установленные на компьютере в настоящее время, и с них можно переустановить систему или установить один и тот же набор пакетов на разные машины. Если у вас Ubuntu Feisty, то *APTonCD* можно установить через *Synaptic*. В стандартном хранилище Debian Etch этого пакета нет, поэтому возьмите его на http://aptoncd.sourceforge.net и установите на каждый из своих компьютеров командой

```
dselect --install aptoncd_0.1~rc-0ubuntu1_all.deb
```

Проще всего пользоваться программой так: пусть один из компьютеров будет подключен к Интернету, а уже на нем можно нарезать диски для обновления остальных. Сначала запустите программу на подключенном компьютере и выберите Create APTonCD (Создать APTonCD). Программа просканирует систему и разыщет пакеты, находящиеся в **/var/cache/apt/archives**, где содержится все, что вы когда-либо инсталлировали (если, конечно, эту директорию специально не чистили). вам будет представлен полный список пакетов. Удалите из списка все, что вам не нужно (например, чтобы втиснуть все на один диск) и добавьте дополнительные пакеты. *APTonCD* сама добавит необходимые зависимости для каждого пакета, если только вы не запретите это явно. *APTonCD* нарезает CD или DVD и создает столько дисков, сколько необходимо для размещения всех файлов. Нажмите OK, и *APTonCD* сформирует один или несколько ISO-образов, готовых к прожигу на любом записывающем приложении. Программа предложит начать прожиг вскоре после того, как подготовит ISO-образы.

Записав образы на диск(и), поместите их на одну из изолированных машин, запустите *APTonCD* и выберите вкладку Restore (Восстановление). Два первых пункта касаются восстановления системы с CD и в принципе могут понадобиться, но к Вашему вопросу отношения не имеют. Третий пункт добавляет диск к системе в качестве хранилища, которое может быть использовано *apt-get*, *Synaptic* или любым другим диспетчером пакетов для обновления компьютера. Заглянув в **/etc/apt/sources.list** или выбрав Settings –> Repositories (Настройка –> Хранилища) в *Synaptic*, вы увидите, что новый CD был причислен к числу доступных источников ПО. Запустите Update Manager (Диспетчер обновлений) – вы сможете увидеть и применить любое обновление системы. Неплохо бы при каждом обновлении очищать **sources.list**, иначе здесь будет создано несколько записей, по одной для каждого CD. НБ

Шаг за шагом: Репозиторий на диске



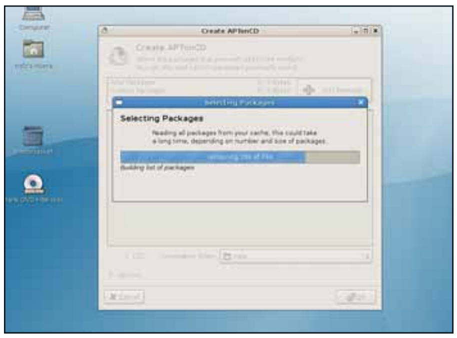
1 Загрузите APTonCD

Пользователи Debian могут загрузить пакет *APTonCD* с http://aptoncd.sourceforge.net/download.html.



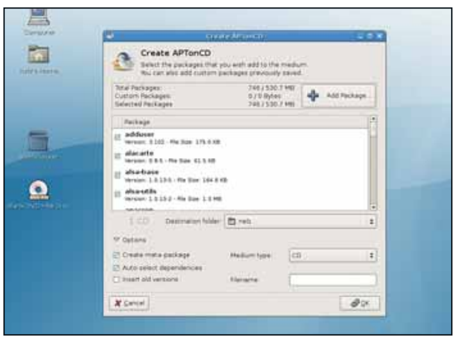
2 Или инсталлируйте его

На Ubuntu (и Kubuntu) все, что нужно для инсталляции – это набрать **aptoncd** в поле поиска Synaptic.



3 Сканирование системы

APTonCD сканирует все архивные Deb-файлы, чтобы составить список пакетов для включения в свое хранилище.



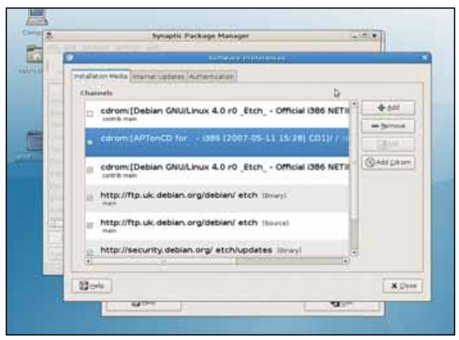
4 Отбор

Добавляйте и удаляйте файлы по своему усмотрению, затем задайте параметры для записи диска. Для добавления пакетов пользуйтесь кнопкой Add Packages (Добавить пакеты) или перетаскивайте файлы в окно.



5 Добавление хранилища

Запуск *APTonCD* на другом компьютере добавит только что созданный диск к числу источников ПО для apt.



6 Завершенный CD

Synaptic отображает новый CD как источник для загрузки обновлений. Не забудьте удалить старые хранилища *APTonCD*, когда они станут не нужны.



Лучшие новинки открытого ПО на планете

LXF HotPicks



Ник Вейч
Чудо-математик, фанат Flickr, иногда редактор, Ник из высокой башни LXF краем глаза осматривает поле свободных программ в поисках хорошеньких

Менеджер анонимности

Tork

Версия 0.17 **Сайт** http://tork.sourceforge.net

Разговоры стоят дешево. Все это знают. А *Tork* вообще бесплатен, значит, он даже лучше. *Tork* – инструмент для настройки и запуска сервиса анонимности Tor. Если вы не знакомы с самим Tor, то вот вам предыстория...

Tor – это суперсеть, расположенная в Интернете. Соединяясь сетевыми тунелями, узлы Tor могут использоваться для распределения пакетного трафика, мешая правительствам, бизнесу и злоумышленникам понять направление передачи ваших данных. Даже сервера сети, используемые в качестве узлов, не имеют понятия, кому предназначаются пакеты – они знают только узел, откуда пришли пакеты, и узел, куда следует их передать. Все пакеты к тому же зашифрованы. Человек, конечно, сумеет засунуть свой нос в вашу деятельность, но работу автоматических систем сбора данных это весьма затрудняет. И отлично, потому что у вас может быть много причин не желать, чтобы ваш провайдер или кто-то другой блокировали трафик или следили за вашими передвижениями по Сети.

А при чем здесь *Tork*? Ну, Tor дело хорошее – для тех, чей любимый вид досуга – возня с файлами настроек и расшифровка их сакрального синтаксиса; но остальные, вероятно, предпочтут установить утилиту-помощника. Являясь KDE-оболочкой, *Tork* поможет вам настроить Tor и запустить анонимный сервис, если вы не хотите постоянно испытывать перегруженность и сложность работы в анонимном режиме.



› **Ваши действия в сети прикроет технология анонимности *Tor*.**

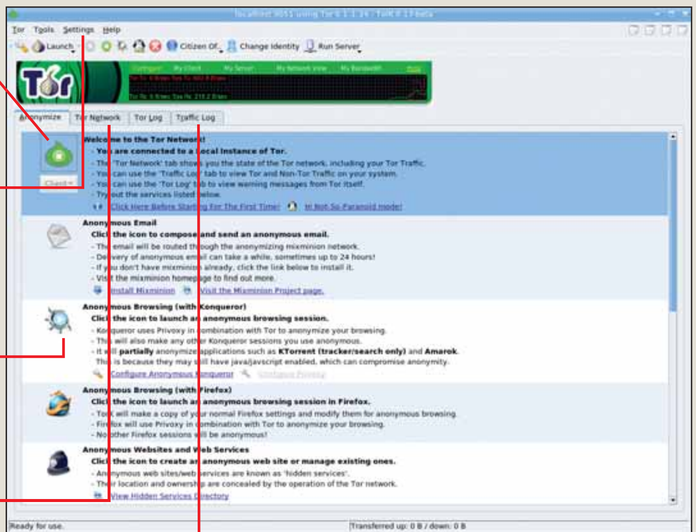
> «Прекрасный способ сделать активность в Интернете анонимной.»

При первом запуске *Tork* вы пройдете через этапы установки, которые настроят службы Tor, задав вам несколько вопросов. Учтите, что для корректного вхождения в сообщество Tor вы должны иметь возможность запустить собственный узел, а значит, иметь общедоступный сетевой адрес (для этого может понадобиться настроить маршрутизатор, изменить настройки сети или умаслить сисадмина). Вы можете запустить и Tor, и *Tork* только как клиент, но не исключено, что для правильной работы вам все же придется менять часть настроек NAT.

После настройки *Tork* на экране отобразится удобный список доступных сервисов, включая SSH, отправку почты, обновление PGP-ключей и тому подобное. В большинстве случаев, предоставляется выбор приложения (например, использовать *Firefox* или *Konqueror*, беседовать при помощи *Kopete* или *Ksirc*), и похоже, что самые популярные клиенты поддерживаются.

Кроме предоставления доступа в Tor-сеть, *Tork* также следит за соединениями и трафиком, ведет журнал использования системы и рисует прелестные графики. Что всегда приятно. Если вы не собираетесь стать полным параноиком, *Tork* – это хороший способ сделать вашу реальную деятельность анонимной, без постоянного обращения к командной строке.

Исследуем интерфейс Tork



Запуск
Щелкните здесь, чтобы приступить к изучению *Tork*.

Настройка
Если вам надо создать или изменить ваши настройки, запустите мастер из этого меню.

Прячься! Прячься!
Просто щелкните на иконке, чтобы запустить выбранный сервис.

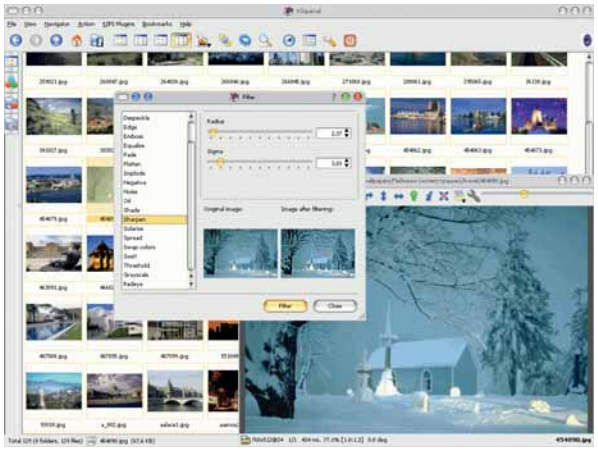
Вид
Просмотр доступных серверов и выполненных соединений.

Монитор
Монитор сети и журналы будут показаны здесь.

Просмотрщик изображений

KSquirrel

Версия 0.7 **Сайт** http://ksquirrel.sourceforge.net



Миру сейчас нужна любовь, нежная любовь. Но вместо нее он получает еще один просмотрщик-каталогизатор в виде *KSquirrel*, с любовью (надо думать) разработанный Дмитрием Барышевым. После запуска вышеупомянутого (*KSquirrel*, а не Барышева), вы окажетесь перед отвратно-ужасной заставкой, но не воротите нос – вообще-то *KSquirrel* весьма приятен. Ну, кроме выворачивающей кишки мерзкой иконки. Ладно, забудем иконку и заставку... ох, да еще и тошнотворную вертящуюся штуковину в правом верхнем углу, которая всего-навсего открывает web-сайт... итак, отвлекаясь от всего этого, *KSquirrel* действительно неплох.

Главное окно похоже на любое другое окно KDE в виде панели-с-полосами прокрутки. Скорее всего, основная часть будет пуста, потому что нужно понажимать на кнопки для перемещения по/к папкам с изображениями, которые вы хотите открыть. А вот и козырь *KSquirrel*: его библиотеки понимают массу графических форматов, включая экзотические, вроде древних графических файлов Commodore 64 и Atari ST (хотя имеются и более полезные добавки, например, PSD от Adobe Photoshop). Пока отсутствует поддержка RAW-изображений, а значит, увы, нет поддержки множества проприетарных форматов для цифровых камер.

Несколько иконок в левой части панели отвечают за полезные навигационные точки, а также предоставляют доступ к разделу категорий *KSquirrel*, где вы можете создавать каталоги и переносить в них изображения, относя их к некой категории. К сожалению, документация пока не очень обильна – ее скорее нет, чем она есть – так что трудно понять, для чего эта информация о категории нужна. Аналогично, имеется корзина изображений, которая, судя по названию, должна выполнять некие действия, но их как-то не обнаруживается. Зато опция слайд-шоу хороша, и работает прекрасно.

› Управляйте вашими файлами изображений проще. И смотрите слайд-шоу.

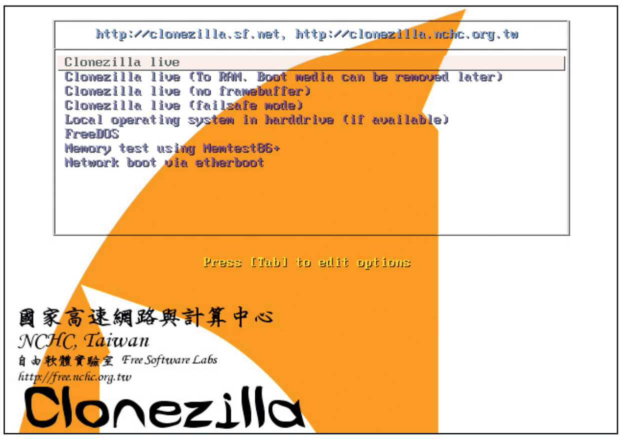
> «Его библиотеки понимают множествографических форматов.»

Исходный код, как всегда, на DVD. Прежде чем перейти к основному приложению, вы, естественно, должны будете собрать *ksquirrel-libs*. Если на этапе *configure* возникнут проблемы с обнаружением библиотек, укажите их местоположение. Например, файл **pkg-config** может оказаться в вашем дистрибутиве не там, где нужно. Удачи!

Клонирование разделов

Clonezilla

Версия 1.0.3 **Сайт** http://clonezilla.sourceforge.net



Чего вы боитесь больше всего? Если честно? Не того ли, что когда-нибудь, возможно пока вы спите, все ваши данные и файлы, которые вы когда-либо создавали, заполняли, исправляли, каталогизировали и сохраняли – все приложения, которые вы, намучавшись, довели до ума – внезапно и необъяснимо исчезнут с пластин вашего злополучного диска?

Так и есть? Что ж, весьма печально. Поверьте нам, есть вещи и похуже. Но потеря всех данных – большая неприятность, и потому мы, компьютерщики, любим тешить себя тем, что у нас есть резервные копии (в смысле, несколько неподписанных DVD, разбросанных среди нижнего белья).

Иногда резервирования ваших файлов и данных недостаточно – если вы хотите иметь возможность молниеносно восстановить работоспособную систему, то вам нужен полный дамп разделов диска, о чем мы и поговорим. Это тем более полезно, если в вашем подчинении парк компьютеров, на которых придурки-пользователи регулярно гробят всю ОС; разве не хотели бы вы всегда иметь возможность отката к настроенной чистой системе?

Рынок домашних пользователей, вероятно, полностью захвачен *Partition Image*. *Clonezilla* скорее ориентируется на системных администраторов или людей, обслуживающих множество систем, и специально предусматривает поддержку группового сетевого клонирования. Фокус тут в возможности работать с несколькими машинами одновременно. Автор протестировал *Clonezilla*, клонируя одновременно 41 машину, и обнаружил, что образ в 5.6 ГБ разворачивается менее чем за 10 минут!

Типы разделов не ограничены Linux: *Clonezilla* поддерживает множество известных файловых систем Unix (ext2, ext3, reiserfs, xfs, jfs), но также работает с FAT- и NTFS-разделами ОС Windows.

> «Хотели бы вы всегда иметь возможность отката к чистой системе?»

› Заставка не блещет красотой, но зато здесь рассортированы ваши резервные копии разделов.

Clonezilla базируется и требует для своей работы несколько компонентов, например, *Partition Image* и *Ntfsclone*. Он использует сервер DRBL для отправки образов к клиентам с загрузкой по сети и, если честно, требует некоторой настройки – настоятельно рекомендуем посетить сайт программы, для дополнительной информации по установке и конфигурированию. А мы прилагаем ISO-образ Live CD *Clonezilla*, он поможет вам в повседневном клонировании настольных ПК.

IDE для Python

Eric

Версия 4.0 **Сайт** www.die-offenbachs.de/eric/index.html

Прекрасный инструментарий разработки – это одно, но интегрированная среда разработки (IDE) действительно может собрать все вместе и здорово облегчить вам жизнь. Для Mono есть *MonoDevelop*, для KDE – *KDevelop*, а для Python есть *PyDevelop* – хм, нет, извините, для Python есть *Eric*.

Вы можете считать, что коль скоро Python – скриптовый язык, то полная IDE вам ни к чему, и будете совершенно правы: вам она не нужна! Достаточно текстового редактора. Наберите весь свой код в одну строку, да и выполняйте его. Но наличие IDE намного все упрощает.

Базируясь на инструментарии *Qt*, *Eric* обладает всеми функциями, присущими высококлассной IDE, наряду с обычными удобствами: подсветкой синтаксиса и парных скобок, сворачиванием кода, множественными видами и т.д. Да, тут налицо побольше, чем просто классный текстовый редактор. *Eric* предоставляет всевозможные инструменты и надстройки для облегчения написания Python-кода. Распознается даже то, что вы хотите вести разработку на нескольких языках (особенно C). Вы обнаружите инструмент рефакторинга, автозавершение для созданных вами классов, макросы, интерактивную оболочку и... нет, вы просто обязаны попробовать.

Начинаем игру

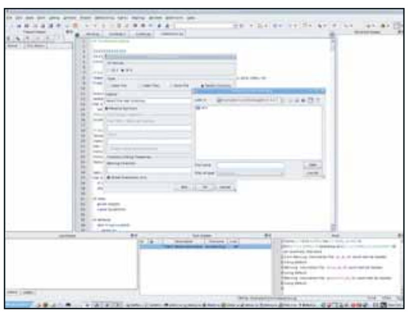
Установка может быть не без проблем, потому что *Eric 4* использует самые последние доступные версии программ, и весьма вероятно, что понадобится собрать из исходных текстов еще и зависимости. На ваше счастье, мы такие добрые, что поместили справа полное руководство по установке. И нечего скулить.

Проекты можно создавать локально или загружать из системы управления версиями *CVS/Subversion*. Доступных режимов множество, и вы наверняка захотите поэкспериментировать с файлами конфигурации для изменения стандартного расположения окон – вид у IDE не совсем обычный: здесь больше экранных элементов по умолчанию, включая интерактивную оболочку (весьма полезную).

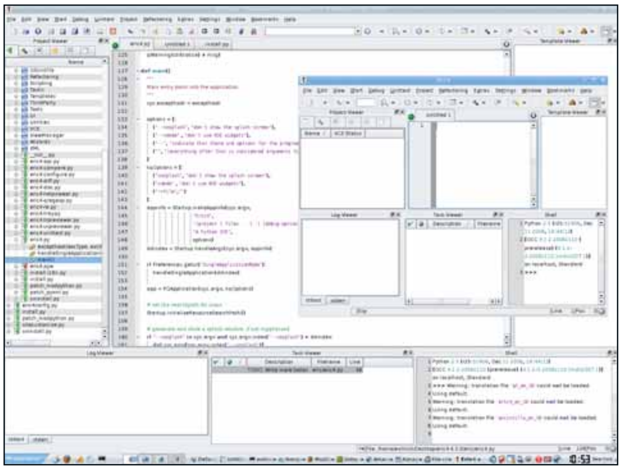
Окно внизу посредине содержит список задач, его можно генерировать автоматически из TODO-комментариев в коде. Когда всего этого станет слишком много, настройте фильтр, чтобы оставить на экране только самое необходимое.

Если вы хотите создать для вашего приложения графический интерфейс *Qt*, можете уже запускать инсталлятор, а для задач полегче мастера соберут вместе простые окна выбора файлов или диалоги за несколько щелчков мышью.

Да, простой скрипт вполне можно настучать в текстовом редакторе, но мы думаем, что *Eric* – это способ создания более требовательных проектов.



› Мастер диалогов Qt быстро разделается с проблемами открытия файлов.



› Интерфейс *Eric* может показаться непривычным для IDE, но все, что нужно, тут на видном месте – а не нравится, так реформируйте его.

> «Предоставляет все средства для удобного создания кода Python.»

Установим и настроим

Eric, для корректной работы, требует несколько сторонних библиотек Python, и пока что некоторые из них вряд ли доступны в виде пакета для вашего дистрибутива. Исходные тексты выложены на LXFDVD, но в любом случае, вам следует заглянуть на сайты для получения новых версий. Вот что вам нужно, в порядке установки:

» python-devel

Если вы не собирали Python из исходных текстов, то вам нужны **devel** или **dev** пакеты для вашего дистрибутива – заголовочные файлы необходимы *Sip*.

» Sip

Sip (www.riverbankcomputing.co.uk/sip/download.php) – это инструмент для Python, который облегчает создание привязок для библиотек, написанных на C или C++. Загрузите и распакуйте исходные тексты, как обычно, затем войдите в каталог и выполните следующие команды:

```
python configure.py
make
make install
```

» QScintilla

Scintilla – хорошо известная и активно используемая библиотека-инструмент для редактирования текста. Как вы догадались, *QScintilla* (www.riverbankcomputing.co.uk/qscintilla/download.php) – это ее *Qt*-порт, используемый *Eric* для основной функциональности редактирования.

Распакуйте архив, набрав следующее:

```
cd Qt4
qmake QScintilla.pro
make
make install
```

Если произошла ошибка компиляции, то, вероятно, потому, что у вас установлен и *Qt3*, и *Qt4*, и используется не та версия *qmake*. Возможно, вам придется указать полный путь к *qmake*, типа:

```
/usr/lib/qt4/bin/qmake
```

» PyQt

Это привязки Python для библиотек *Qt*, полезные для любых приложений, но особенно для *Eric*. И вновь вам потребуется последняя версия, которая, вероятно, пока что отсутствует в репозитории вашего дистрибутива. Зайдите на www.riverbankcomputing.co.uk/pyQt/download.php. Распакуйте архив и перейдите в полученный каталог. Если у вас были проблемы с *qmake*, можете использовать ключ **-q** для указания его расположения, как показано ниже:

```
python configure.py -q /usr/lib/qt4/bin/qmake
make
make install
```

» Привязки QScintilla

Ха-ха! Это трюк. У вас уже есть нужный код. Теперь, когда *PyQt* установлен, вы можете создать Python-привязки для него. Вновь вернитесь в каталог с исходными текстами и соберите их так:

```
cd Python
python configure.py
make
make install
```

» Eric

Проделав все предыдущее, вы наконец-то можете установить сам *Eric*! К счастью, это намного проще, чем предыдущие шаги. Выполните

```
python install.py
```

и все!

Вот так все просто. Теперь запустите IDE, набрав *eric4* в командной строке.

HotGames Развлекательные приложения

Гонка с препятствиями

Sturmbahnfahrer

Версия 14 **Сайт** http://www.sturmbahnfahrer.com

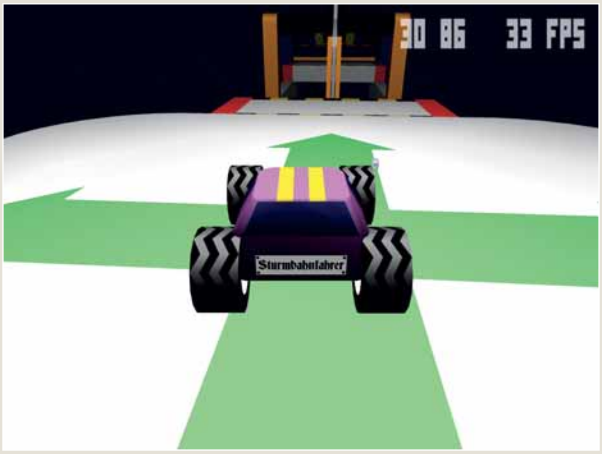
С тех пор, как мир узрел изысканную жуть *Stunt Car Racer*, поколения восторгались 3D-играми-гонками, где машины врезались в препятствия. Ну может, не полные поколения, поскольку *SCR* вышла в 1989. Но в компьютерный век этого времени явно хватит, чтобы завести пару ребятишек и показать им клавиши WASD.

Но мы отвлеклись. Оставим *Stunt Car Racer* и займемся *Sturmbahnfahrer*, игрой-триллером сегодняшнего дня по части автомобильных аварий – конечно, в ремнях безопасности прекрасной 3D-игры, без риска больших страховых выплат или привлечения внимания автоинспекторов.

Запустив *Sturmbahnfahrer*, вы сперва знакомитесь со своей машиной (монстром-драгстером [разновидность гоночного автомобиля, – прим. пер.]), затем, взревев двигателем, едете и исследуете арену действий, ради ознакомления с препятствиями. Среди препятствий – трамплины, ужасные шипы, двигающиеся платформы, башни и опрокидывающиеся ящики (имелись в виду ящики для фруктов, но не будем судить авторов) и разные другие ловушки и фокусы.

Вероятно, вы обнаружите, что не худо бы отдохнуть от мысли все время быть крутым – здесь работают вполне реальные, а иногда и преувеличенные, законы физики, и если вы воображаете, что крутой поворот удастся на любой скорости, то, вероятно, итогом будет проверка на прочность кувырком катящейся клетки.

Большое трюкачество напрямую связано с расчетом времени и мастерством. Не имеет смысла разгоняться на трамплине и нажимать клавишу Действия – скорее всего это приведет к выполнению пары кульбитов и остановке на вашей крыше. Вам также следует аккуратно управлять машиной, а иногда даже использовать тормоза.



› А вы-то думали, что трудность – это добиться приглашения на выпускной бал в Кремле.

Полоса препятствий – это, в принципе... ну, мы не хотим сказать «невозможно», но до ее конца мы ни разу не добрались. Если вы сумеете-таки показать впечатляющее время, то ваш долг – написать нам об этом!

Аркада

TaxiPilot

Версия 0.9 **Сайт** http://taxipilot.sourceforge.net

Представьте себе Commodore 64. Вообразите глупую, но захватывающую игру. Представьте такси. Но в пространстве.

Вот и все ментальные усилия, что вам придется предпринять для подготовки к этому букету развлечений. *TaxiPilot* – это еще одна из тех игр, где вы доставляете персону X из пункта А в пункт Б или В. Вам следует сделать это быстро, иначе вы не получите чаевых. И вы должны сделать это, не врезаясь во что-либо, потому что это может повредить ваше такси и расстроить пассажиров. Да следите, чтоб не кончился бензин. Это будет невезение. Есть еще что-нибудь?

Удивительно, сколько удовольствия можно получить от такой скучной идеи. Ребекка из команды **LXF**, сказала, что ее отец – таксист, и она считает, что игра попала в точку. Только не по части пространства. Если вы зарабатываете на жизнь вождением такси, то поначалу не заметите ничего особенного, но мы готовы поспорить, что ваше такси не имеет регуляторов высоты. Мы также готовы побиться об заклад, что ваше такси не может передвигаться между планетами, астероидами, сброшенными контейнерами *Roswell* и другими космическими объектами. Если мы ошибаемся, и оно это умеет – свяжитесь с нами.

Да, концепция *TaxiPilot* – доставка пассажиров из одного места в другое по их желанию – проста (хотя половине таксистов в Нью-Йорке ее никак не вдолбить).

Но *TaxiPilot* на этом не останавливается: игра очень расширяема. Держите счетчик включенным, пока ищете одну из множества доступных миссий (некоторые используют вообще другую графику), а если вам стало очень скучно, или вы просто хороший человек, желающий доставить людям радость, то совсем просто создать свою собственную.



› Сколько вы обычно платите за проезд через рощу смерти?

Эта новая версия потребовала двух лет разработки, но в ней есть несколько новых особенностей и два новых уровня. Релиз нуждается в KDE и во всем сопутствующем стандартном инструментарии, и хотя это бета-версия, она выглядит вполне законченной, так что не видно препятствий для установки и игры. Когда надоест выигрывать, вспомните, что разработчик Томас Фридрихсмайер [Thomas Friedrichsmeier] ищет помощников – посетите его домашнюю страничку и вливайтесь в проект.

Редактор 3D-моделей

Misfit Model 3d

Версия 1.2.4 **Сайт** http://www.misfitcode.com/misfitmodel3d

Blender очень хорош, если у вас есть проницательность, чтобы суметь изучить его. Как и многие мощные приложения, он требует для освоения определенного усердия. А иногда *Blender* – просто перебор для ваших простых задач. К сожалению, по части интерфейса пользователя 3D-пакеты все еще весьма сложны. В отличие от разработчиков текстовых процессоров и web-браузеров, создатели 3D-приложений все еще не пришли к соглашению, как должен выглядеть интерфейс пользователя. Для пользователя имеется три различных пути изучения таких приложений (потому что они очень отличаются), и наработанный опыт часто не годится для другой программы. Но вообще-то, много ли надо, чтобы наскоро слепить 3D-объект для web или подлинно кровавую деталь для вашего последнего побоища в Quake?

Misfit Model 3d, или *MM3D*, не столь труден для понимания, и к тому же может использоваться для некоторых весьма сложных задач. Для отрисовки интерфейса пользователя в *MM3D* используется *Qt* и OpenGL, хотя и по минимуму, что приятно: вряд ли вам захочется постоянно отвлекаться от процесса создания моделей. Автор Кевин Вустер [Kevin Worcester] говорит: «*MM3D* создан быть легким в использовании и легким для расширения, при помощи дополнительных модулей и скриптов».

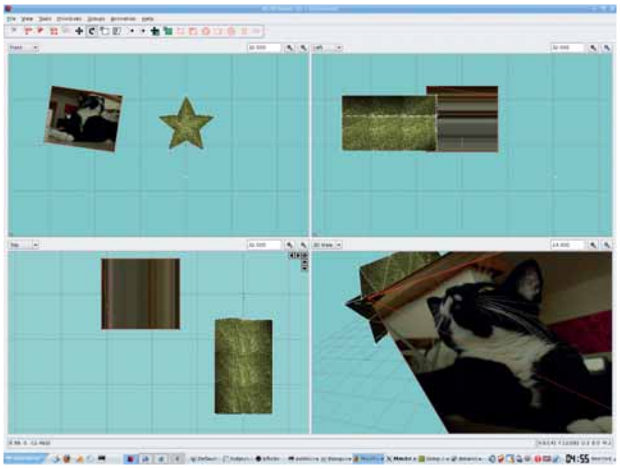
Однако сперва меню может показаться непонятным, и если честно, его следует почистить, поскольку не всегда очевидно, где что следует искать. К счастью, нет нужды постоянно использовать меню, поскольку все часто используемые инструменты расположены на панели инструментов. Первый набор элементов управляет выбором объектов/вершин, имеется несколько инструментов трансформирования, и рамками обведены доступные примитивы. Не ищите блестящих NURBS-режимов моделирования или чего-то подобного – это простой полигональный редактор моделей.

Добавить примитив легко. Щелкните на желаемом типе, затем просто нарисуйте его в одном из имеющихся экранов ортогональных проекций (сверху, сбоку, спереди).

Искусство импорта

Добавить текстуру также легко – понадобится только растровое изображение. Скоренько изготовьте шедевр в GIMP, сохраните в подходящем формате (сгодятся JPEG или PNG), а затем используйте инструмент меню Groups [Группы] в *Misfit Model 3d* для его загрузки. Меню Groups также необходимо для определения групп выделенных объектов, что облегчает текстурирование.

Панель 3D-вида может быть отрисована текстурами и различными типами теней, чтобы натолкнуть вас на лучшие идеи о том, что должно получиться в итоге. Эта часть использует OpenGL – если вы хотите, чтобы она выглядела хорошо, то позаботьтесь о видеокарте с 3D-ускорителем и правильно настроенных под нее драйверах OpenGL. Например, если у вас видеокарта от NVidia, то вам обязательно нужны официальные драйвера NVidia, к сожалению не с открытым кодом. Приложение будет работать и без 3D-ускорения, но вы обнаружите, что изображения рисуются и обновляются очень медленно.



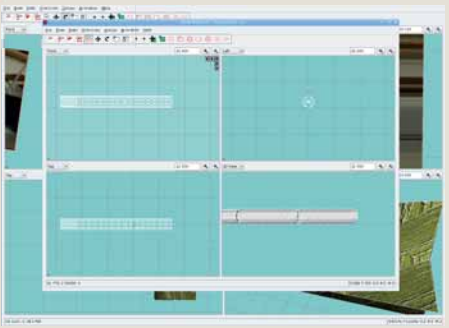
› Перестройка Вселенной, по вершине за раз.

Поддержка файлов в *MM3D* не блещет, но *MM3D* понимает популярные форматы, такие как *MilkShape*, *Quake MD2* и вездесущий Wavefront OBJ, и если вы не сможете сохранить файл сразу в желаемом формате, то, понажимав клавиши, добьетесь нужного результата в несколько приемов. Однако полезно не забывать сохранять все рабочие файлы также и в «родном» формате, поскольку некоторые настройки могут потеряться при сохранении, например, в формате .OBJ.

Не обольщайтесь, данное приложение не конкурент *3D Studio Max* или *LightWave*. И не собирается им стать: это лишь простой и быстрый способ создания простых, малополигонных моделей, которые можно использовать в играх или где-то еще.

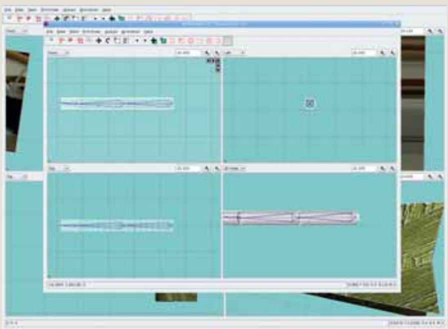
> «В Misfit Model 3d добавить текстуры просто: нужно лишь растровое изображение.»

Шаг за шагом: Создание скелетной анимации



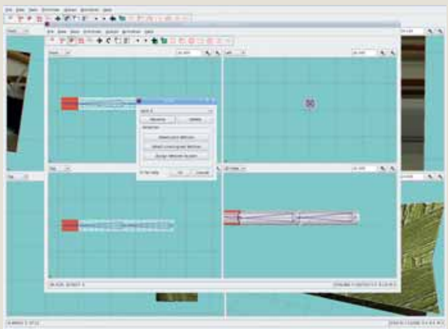
» Создаем модель

Создайте три цилиндра, лежащих на одной линии. Вам не нужно изменять их стандартные параметры, но убедитесь, что вы видите, где заканчивается один цилиндр и начинается следующий!



» Добавляем кости

Выберите Tools–>Create Bone Joint [Инструменты –> Создать суставы]. Теперь щелкните один раз в начале линии цилиндров, и один раз в конце. Создастся линия костей. При помощи Select Connected Mesh Tool [Инструмента выбора соединенных объектов] выберите цилиндры.



» Соединяем и анимируем!

Выберите Primitives –> Joints [Примитивы –> Сочленения] чтобы добавить конец кости к соответствующему цилиндру. В режиме Animation –> Start Animation [Анимация –> Начать анимацию] щелкните на Skeletal [Скелетная]. Теперь вы можете двигать суставы модели.

Графопостроитель

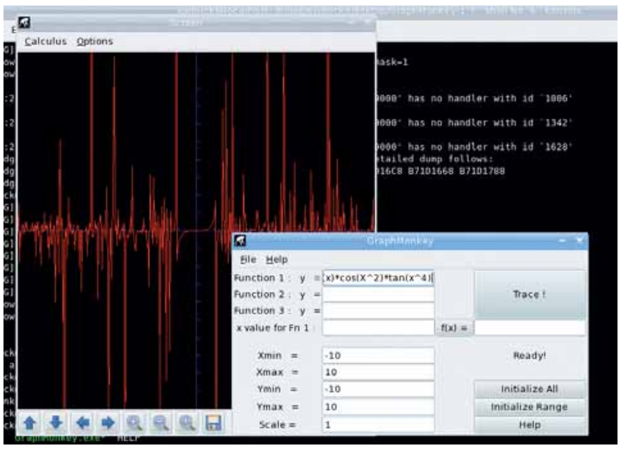
GraphMonkey

Версия 1.7 **Сайт** http://graphmonkey.sourceforge.net

Последний выпуск **HotPicks**, где мы восхваляли плюсы калькулятора *Speedcrunch*, вызвал полемику. Вероятно, это потому, что в тексте мы пренебрежительно говорили о графических калькуляторах. Мы понимаем, что не каждый может представить мысленно вид любой функции. Мы также понимаем, что для таких людей инструменты построения графиков очень полезны.

Успокойтесь, плакальщики и клеветники. В этом месяце мы представляем графический калькулятор, но без калькулятора! Единственная задача *GraphMonkey* – это рисовать графики функций (числом до трех) различными цветами. Поражайтесь изгибам x-x^2. Благоговейно созерцайте все великолепие sin(x)*cos(x^2)*tan(x^4). Просто наберите выражение и нажмите Trace для праздника наглядности. Если вы не знаете, что можно сделать, нажмите Help для вывода списка поддерживаемых функций. Вы можете также ввести примерные значения X для первого выражения и получить вычисленное *GraphMonkey* значение функции в этой точке. *GraphMonkey* умеет также выполнять простые операции над функциями.

Приложение написано для Mono, так что вам потребуется установленная среда Mono. Если у вас проблемы с компиляцией из исходных текстов, просто возьмите бинарный пакет с сайта.



› Я всегда знал, что кривая вывезет…

Архиватор

PeaZip

Версия 1.8 **Сайт** http://peazip.sourceforge.net

Иногда так хочется взять что-то и запихнуть в наименьшее возможное пространство, до тех пор, пока оно вам снова не понадобится... Но хватит о Майке. *PeaZip* – это инструмент для упаковки и распаковки файловых архивов файлов в архивах: картинки, исходные тексты, требования выкупа или что-то еще. Он не столь распространен, как, скажем, *File Roller*, но *PeaZip* имеет что предложить.

Главное из этого – поддержка огромного числа форматов, включая загадочные файлы Windows, которые иногда полезны пользователям Linux. Он также предлагает свой собственный формат (ей-Богу), называемый (сможете догадаться?) PEA, фишка которого – очень мощное шифрование для защищенных архивов.

Простое отображение содержимого архивов достаточно функционально (хотя немного хромает при открытии TGZ файлов: сначала вы просто видите каталог с tar-файлом внутри). Интерфейс пользователя основан на *GTK*, и пару раз показался нам странным, но, вероятно, это проблемы с темами.

PeaZip работает и под Windows, что может пригодиться, если вы собираетесь совместно использовать файлы. И хотя приложения вроде *Konqueror* не поддерживают его напрямую, довольно просто заставить их сделать это – командная строка *PeaZip* имеет простые параметры действий, для безболезненного написания скриптов. LXF

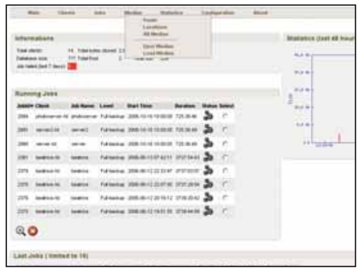
› Делайте большое маленьким – и наоборот.



Также выпущены

Новые и обновленные программы, заслуживающие вашего внимания...

» **Bacula 2.1.12** Инструмент резервирования и восстановления. В этой версии были добавлены функции в администраторскую часть www.bacula.org



› Создавайте резервные копии!

» **Karoshi 5.1.3** Любимый школьный проект *Linux Format* – серверная ОС для облегчения администрирования школьных сетей www.karoshi.org.uk

» **Gnash 0.8 Flash**, но не проприетарный, этот проигрыватель с открытым кодом работает как расширение для популярных браузеров и теперь поддерживает YouTube www.gnu.org/software/gnash

» **KOffice 1.6.3** Еще один офисный пакет для Linux выполнил необходимое исправление множества ошибок www.koffice.org

» **GnuAccounting 0.4.0** Бухгалтерское приложение – с ним не повеселишься, но зато упорядочишь свои средства http://gnuaccounting.sf.net

» **ROX-Filer 2.6.1** Популярный файл-менеджер с множеством продвинутых опций и новой функцией миниатюр http://rox.sourceforge.net/desktop/ROX-Filer



› Файл-менеджер приятной наружности.

» **SafeBox 3.37** Хранит персональную информацию надежно и в безопасности на USB-брелке www.connectina.com/safebox/index.html

» **Webmin 1.350** Лучший инструмент администрирования на базе web – на этот раз добавлено много настроек, улучшений и заплаток www.webmin.com

» **Xine 1.1.7** Новая поддержка *libdca* и множество исправлений для этого почтенного медиа-инструмента www.xinehq.de



LXF DVD95

В этом месяце – DVD с двойной загрузкой и двумя отличными дистрибутивами Linux.



Майк Сондерс любовно подбирает содержимое диска Linux Format, а также поддерживает сайт www.linuxformat.co.uk.

Первые впечатления

Я всегда стремился быть этакой квинтэссенцией компьютерщика, ставя технологию и программирование выше всяких косметических штучек и прикольных эффектов на рабочем столе. Однако с появлением Fedora 7, кажется, я начинаю менять свое мнение. Она такая… красивая. Уж не знаю, что общего у Linux и воздушных шариков, но их использование – весьма оригинальная идея, и они куда приятнее для глаза, чем случайные цветовые завихрения.

Многие ворчали на команду Ubuntu за выбор для своего дистрибутива темных коричневых тонов, но они привлекают внимание – именно потому, что весьма необычны. Это нам и нужно в мире Linux: мы уже наелись тем в стиле Windows, вызывающих разве что недоумение новичков. У нас есть суперневразумительные менеджеры окон, подобные *Wmii*, специально для продвинутых пользователей. Но на деле нам нужен рабочий стол Linux, с характерными чертами, которые сделали бы его таким же узнаваемым, как у Microsoft и Apple.

Большинство популярных дистрибутивов поддерживают раскладку Gnome и KDE по умолчанию, что очень полезно, и было бы неплохо иметь стандартный набор значков рабочего стола Linux (Tango?). В Windows, при всех проблемах этой ОС, пользователи по крайней мере могут переходить с машины на машину, не опасаясь, что не обнаружат на привычном месте панель задач и панель управления. Нам нужно, чтобы и в Linux было что-то подобное – настраиваемый интерфейс, да при этом с простой навигацией, каким бы дистрибутивом вы ни пользовались.

В любом случае, я надеюсь, вам понравится DVD этого месяца: потрясающий экземпляр с двумя полными дистрибутивами. Он содержит инструменты разработки, приложения рабочего стола, игры и другие программы, а также несколько файлов в формате PDF из предыдущих выпусков. Если у вас есть комментарии или предложения по программам, сообщайте!

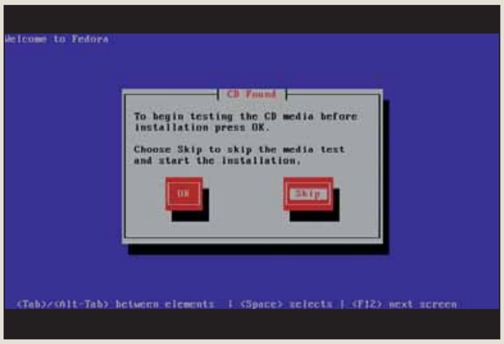
mike.saunders@futurenet.co.uk

Шаг за шагом: Установка Fedora 7



1 Загрузка

Загрузите компьютер, установив в него **LXFDVD**, и нажмите Enter в меню загрузки.



2 Решите, нужен ли тест

Вам предложат проверить диск на наличие ошибок – вы можете спокойно пропустить эту опцию, если только вам уже не встретились ошибки.



5 Разбивка диска

Выберите Create Custom Layout, чтобы вручную настроить разделы на жестком диске. Вам потребуется не менее 10 ГБ на /(root) и 512 МБ для раздела подкачки [swap partition].



6 Выбор программ

Anaconda настроит ваше сетевое устройство. Затем вы можете принять программы по умолчанию, или нажать Customize Now, чтобы сделать свой выбор.

Дистрибутив

Fedora 7

Нет, это не опечатка – это не Fedora Core 7. Слово Core исчезло, и больше нет разделения между репозиториями Core и Extras. В результате базовый дистрибутив Fedora усох в размере (хотя по-прежнему имеет внушительный объем в 2.8 ГБ). Fedora шагала вперед семимильными шагами со времен своих ранних релизов, над которыми многие насмехались, считая их тестовым полигоном для бизнес-линейки Red Hat. Однако Red Hat увязла в своем проекте и пригласила к участию сообщество. Сегодня Fedora – превосходный всеобъемлющий дистрибутив с массой превосходных пакетов программ.

Итак, что же нового в этом релизе?

» **Gnome 2.18 и KDE 3.5.6** Рабочий стол по умолчанию по-прежнему Gnome, но и KDE не обойден хорошей поддержкой.

» **Быстрое переключение между пользователями** С помощью кнопки входа в систему, подобной OS X, переключение на другую учетную запись – секундное дело.

» **NetworkManager Как и в Ubuntu**, эта утилита позволяет вам переключаться между сетями с помощью всего нескольких щелчков мышью – что особенно ценно, если вы используете беспроводное соединение на ноутбуке.

» **SELinux** Эта программа безопасности обзавелась новым графическим инструментом для тонкой настройки.

» **Тема Flying High (Высокий Полет)** Голубой по-прежнему остается цветом Fedora, но новая тема щеголяет гроздьями воздушных шариков.

» **Быстрое управление пакетами** Оно действительно стало проворнее.

Поскольку это универсальный дистрибутив, то при основном уклоне развития в сторону настольных ПК имеется также достаточное количество инструментов разработки и пакетов, ориентированных на сервер. Для установки потребуется x86 ПК с минимум 128 МБ ОЗУ; для графической установки необходимо 256 МБ ОЗУ. Рекомендованная частота процессора зависит от вида решаемых дистрибутивом задач – для базового файлового сервера будет достаточно 300 МГц, но для настольного компьютера лучше иметь 800 МГц или выше. Заметьте, что эта версия Fedora 7 хотя и 32-битная, но с успехом может установиться и работать на 64-битном ПК.

Можете начать с загрузки своего компьютера с **LXFDVD** и нажать Enter в меню загрузки. (Если у вас менее 256 МБ ОЗУ, выберите в нем опцию установки в текстовом режиме) Инсталлятор Fedora, *Anaconda*, очень прост в использовании – см. руководство по установке ниже. Закончив установку и приступив к работе, можете получить дополнительную информацию и подсказки по дистрибутиву на сайте проекта – http://fedoraproject.org, или, если нужна помощь пользователей Fedora, зайдите в раздел Red Hat Linux, Fedora Core, ASP Linux нашего форума на www.linuxforum.ru.



› **Gnome 2.18.0, *Firefox 2.0* и *OpenOffice.org 2.2* – основные программы для настольных ПК в Fedora 7.**



3 Запуск
Программа установки *Anaconda* запустится и предложит вам выбрать язык и раскладку клавиатуры.



4 Дисковые предпочтения
Вы выбираете установку или обновление, и затем указываете, как использовать жесткий диск.



7 Копирование
Программа установки начнет копировать файлы дистрибутива на жесткий диск. Это может занять около получаса.



8 Конечная настройка
Fedora перезагрузится (извлеките DVD), и затем произведет окончательную настройку. Можете создать учетную запись пользователя и войти в систему!

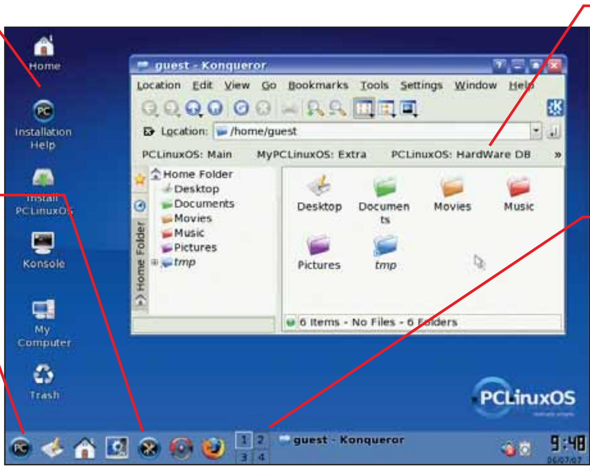
Дистрибутив

PCLinuxOS 2007

Наш второй дистрибутив в этом выпуске – PCLinuxOS, чья популярность за последние 12 месяцев взмыла как ракета. Подобный успех был достигнут благодаря сочетанию стабильной разработки, умелого продвижения и хорошему выбору пакетов по умолчанию – этот релиз включает в себя поддержку Java и Flash, и вам не придется возиться с онлайн-репозиториями после установки, чтобы украсить общение с Web (если вы такой же фанат YouTube, как и мы). В нем множество полезных программ вроде web-браузера *Firefox* и офисного пакета *OpenOffice.org*, так что он является универсальным рабочим столом Linux прямо из коробки.

С PCLinuxOS стоит познакомиться еще и потому, что это Live-дистрибутив: вам не обязательно устанавливать его на жесткий диск. Поясним для тех, кто раньше никогда не использовал Linux и только начал читать наш журнал: это означает, что вы можете начать знакомство с Linux прямо с DVD, не затрагивая свою основную ОС. Просто загрузитесь с диска, а когда наработаетесь, перезагрузите компьютер. Установленным программам Windows ничего не грозит: PCLinuxOS к ним даже не притронется.

Исследуем рабочий стол PCLinuxOS



Рабочий стол
Здесь вы увидите программу установки на жесткий диск и ссылку для получения справки по этой процедуре.

Центр Управления
Нажмите на этот значок, чтобы вызвать мощный инструмент настройки системы PCLinuxOS.

Главное меню
Нажмите кнопку PC, чтобы запустить программы или выйти из системы (выключить компьютер).

Konqueror
Этот файловый менеджер дает доступ к вашим личным файлам, а также используется как web-браузер.

Рабочее пространство
Эти четыре кнопки позволяют переключаться между четырьмя рабочими столами, чтобы было больше места под открытые приложения.



› **Нажмите на значок с отверткой и гаечным ключом и настраивайте все аспекты PCLinuxOS.**

Для бывалых линуксоидов польза Live-дистрибутивов, подобных PCLinuxOS, огромна: в сочетании с USB-брелком, у вас есть портативный Linux, который вы можете запустить на любой машине. Если вам срочно надо поработать, а доступ имеется только к Windows-компьютеру, просто загрузитесь с DVD и сохраняйте созданные файлы на USB-брелке.

Live-дистрибутивы – это также превосходный способ демонстрации Linux: вы покажете друзьям и сотрудникам, что в нем замечательного, и как он работает, не рискуя причинить какой-либо ущерб их компьютерам. Вдобавок Live-дистрибутивы удобны для проверки совместимости оборудования. К примеру, если ваш приятель обзавелся шикарным новым ноутбуком, а вы захотели такой же, можете проверить, хорошо ли он будет работать в Linux. Если же вы полюбите PCLinuxOS, вы сможете установить его на жесткий диск с помощью утилиты Install PCLinuxOS на рабочем столе: она проведет вас через несколько подготовительных стадий, типа создания учетных записей пользователей и настройки программы загрузки, и следовать ее указаниям легко. Версия PCLinuxOS на нашем DVD предназначена для 32-битных x86 ПК, однако же он должен работать без всяких проблем на 64-битных системах. Для достойной производительности мы рекомендуем 1 ГГц CPU и 512 МБ ОЗУ; если хотите установить его на жесткий диск, потребуется не менее 10 ГБ свободного пространства, чтобы создать разделы Linux.

Ниже приводится руководство по быстрой установке, а на www.pclinuxos.com вы найдете подсказки и поддержку, в том числе и оживленный форум, где вам всегда помогут решить проблемы.

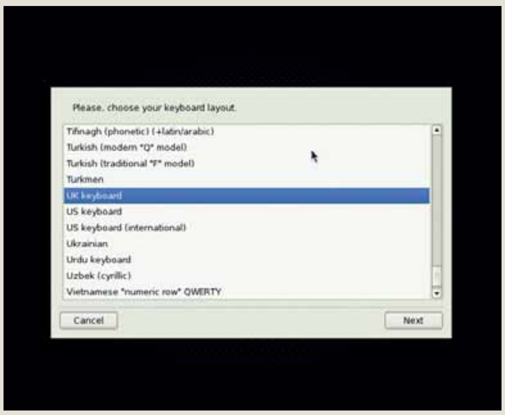
Вход в систему

Войдите в систему под пользователем 'guest' с паролем 'guest'. Администраторский пароль для root – 'root'.

Шаг за шагом: Работа в PCLinuxOS 2007



1 Загрузка
Загрузите компьютер с диска и выберите PCLinuxOS в меню. Если при загрузке дистрибутива возникнут проблемы, попробуйте одну из безопасных опций. Copy To RAM позволяет загрузиться быстро, если у вас более 1 ГБ ОЗУ.



2 Вход в настройки
Вам зададут несколько вопросов о раскладке клавиатуры и о вашем местоположении. Затем вам предложат настроить сеть.

Программное обеспечение

Инструменты разработки

Руки чешутся от желания написать код? У нас есть раздел новинок компьютерного мира как раз для вас, в разделе **Разработка**. Наиболее значимая новость, что GCC, компилятор для Linux, считающийся стандартом, обновлен до версии 4.2.0. Теперь он поддерживает OpenMP для C, C++ и Fortran, что расширяет возможности параллельного программирования.

По части Java, GCC обзавелся новым переключателем -static-libgcj, компонующим библиотеку *libgcj* с исполняемым файлом – это бонус, если вы пытаетесь собрать бинарники, способные работать со всеми возможными дистрибутивами. Еще одно полезное дополнение – набор флагов -mtune=native и -march=native; они используют команду CPUID для определения типа процессора, установленного на компьютере, и вы сможете оптимизировать код, не указывая его вручную.

Разработчики приложений с графическим пользовательским интерфейсом, особенно те, кто работает с программами KDE, должны познакомиться с новым релизом *Qt 4.3*. Инструментарий Trolltech значительно улучшился после версии 4.2; наиболее значимое усовершенствование – включение *QtScript*. Основанный на ECMAScript (известном нам как JavaScript), он соединяется с сигналами *Qt* и механизмом слотов, и позволяет вам добавлять поддержку скриптов к своим приложениям. Среди других новых функций – поколение файлов SVG и более скоростной движок OpenGL.

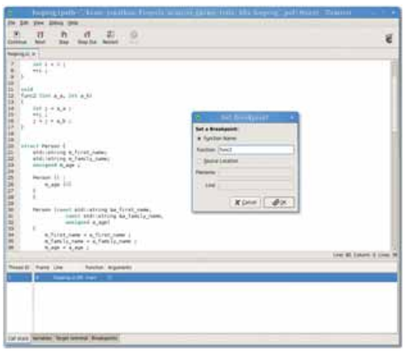
Жизнь бьет ключом в Mono: базовый пакет Mono был обновлен до версии 1.2.4, а *MonoDevelop* добрался до 0.14. Чтобы установить самую последнюю версию Mono, вам придется сначала удалить все старые версии Mono с помощью менеджера пакетов – в противном случае вы рискуете получить смешение двух версий, способное привести к появлению всяких странных ошибок в ваших приложениях.

Удалив Mono, скопируйте файл **mono-1.2.4_4-installer.bin** из раздела **Разработка/Mono** нашего DVD в домашнюю директорию, запустите терминал и введите

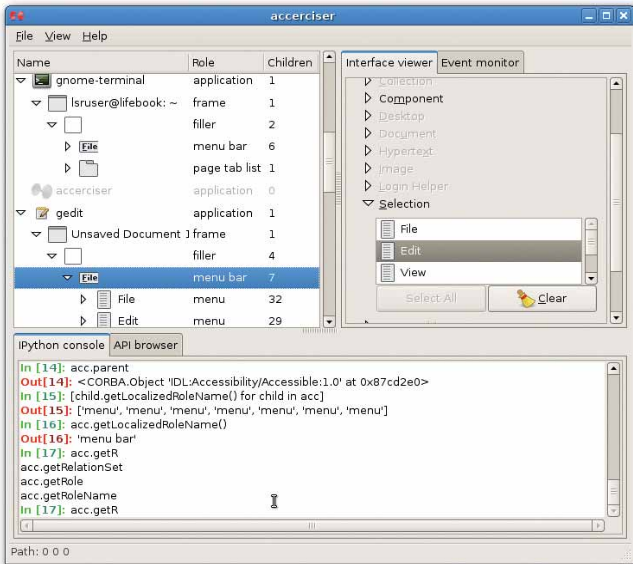
```
chmod +x mono-1.2.4_4-installer.bin
./mono-1.2.4_4-installer.bin
```

Это сделает программу установки Mono исполняемой и запустит ее. Следуйте подсказкам, и в вашей домашней директории появится блестящая новая установка Mono, готовая к использованию. Программа установки даже модифицирует ваши настройки $PATH, чтобы они не путались с предыдущими версиями Mono.

Стоит познакомиться еще с несколькими программами: *Edyuk* – это быстро прогрессирующая IDE для программ на основе *Qt*, а *Nemiver* замахивается на графическую отладку в Gnome. *Konch* – интригующий проект, позволяющий создавать маленькие программы в системном лотке с помощью скриптовых языков, таких как Python или даже *Bash*!



› *Nemiver* подключается к GDB и обеспечивает графический режим отладки в Gnome.



› *Accerciser* помогает делать ваши программы более доступными, проверяя использование подходящих технологий.

Документация

Журнал в формате PDF

Если у вас есть настроение что-то почитать, предлагаем вам три статьи из предыдущих выпусков Linux Format в разделе **Журнал/PDFs**. Вы можете просмотреть их при помощи своей PDF-утилитой рабочего стола – в Gnome это *Evince*, а в KDE воспользуйтесь *Kpdf*. Мы начали со статьи LXF75 про *MythTV* – полного руководства по мощной программе записи персонального видео (она же и медиацентр). *MythTV* печально известен сложностью настройки, но наше руководство объяснит, какое оборудование вам понадобится и как его скомпилировать, и даст различные опции настройки. Также мы покажем, как извлечь максимум из *MythTV* – поставить на паузу прямую трансляцию, просмотреть телепрограмму или сделать запись через сеть.

Следующий материал – анализ женского участия в мире Open Source, опубликованный в LXF75*. Всем нам известно, что в мире Linux доминируют мужчины, однако почему так сложилось? Может быть, причина в дискриминации по половому признаку, или виноват предрассудок, что все программисты – обязательно мужчины? Мы анализируем причины, превратившие Linux в явление преимущественно мужское, и приводим мнение нескольких женщин-профессионалов в мире Open Source.



› Наше руководство поможет вам присоединиться к *GIMP*.

И, наконец, статья LXF82 о разработке *GIMP*. Участие в проекте свободного ПО – будь то программирование, популяризация или документирование – вещь очень благодарная, но часто бывает непонятно, с чего начать. Наше руководство объясняет, как совершенствуется мощный проект *GIMP* и что вы можете сделать, чтобы помочь изгнать ошибки из кода. Мы покажем вам пошаговый процесс исправления ошибок в древе исходных текстов *GIMP*, с последующей сборкой приложения заново с учетом исправленных ошибок. Почему бы не помочь превратить *GIMP 2.4* в лучший релиз на свете!

* – *на английском языке*



3 Вход

Войдите в систему под логином 'guest' и с паролем 'guest'. Закончив работу с PCLinuxOS и выйдя из системы, нажмите здесь Системное Меню, чтобы выключить компьютер.

Другие программы

Новые релизы

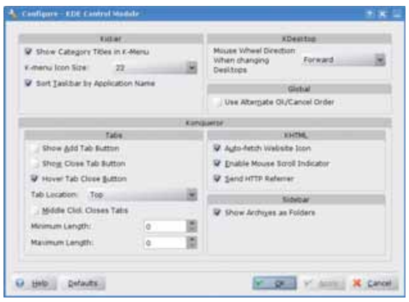
На диске этого месяца – не только дистрибутивы и документация. В сфере разработки программ ведется бурная деятельность, и мы подготовили большое количество программ, чтобы вы могли с ними познакомиться.

В разделе **Рабочий стол** пользователям Gnome стоит обратить внимание на *Wallpapoz* – это умный маленький инструмент, настраивающий вид фона рабочего стола. Большинство из нас в качестве обоев использует статические изображения, в лучшем случае меняя их раз в неделю, а *Wallpapoz* умеет менять фон рабочего стола хоть каждые пять минут, или каждый раз, когда вы переключаетесь на другое рабочее пространство.

Можете даже соединить несколько обоев на рабочем пространстве. Пусть это и не самая важная программа в мире, однако она, без сомнения, помогает освежить ваш рабочий стол (а может быть, и ваше внимание!)

Пользователям KDE мы предлагаем *Tweak*, удобную утилиту для скрытых настроек рабочего стола. Хотя Центр Управления KDE и предоставляет удобные опции для настройки практически всего, все же есть несколько настроек, которые нельзя изменить обычными средствами; для них нужен специальный инструмент. *Tweak* поможет изменить порядок сортировки панели задач, поменять местами кнопки OK и Cancel, реверсировать функцию переключения рабочих столов с помощью колесика мыши и точно настроить множество аспектов *Konqueror* (например, отправку ссылок и добавление кнопок управления вкладками). Если вы – опытный пользователь KDE и придирчивы к мелочам, вам просто нельзя не взглянуть на него – именно он может нанести на ваш рабочий стол тот штрих, по которому вы давно уже страдали.

Далее в разделе **Графика** вы найдете *Fotox*, помогающий улучшить фотографии, сделанные цифровой камерой. Многие из нас любят чистить снимки в *Gimp*, но его роскошь – явный перебор, если надо просто немного



› Некоторые настройки KDE скрыты – но вы доберетесь до них с помощью *Tweak*.

подкорректировать яркость. *Fotox* позволит уменьшить дымку и туман на фото, снизив уровень белого и усилив цвета, да вдобавок способен исправлять передержку и недодержку изображений.

Кроме того, вы можете изменять размер, переворачивать и перетаскивать изображения, и все это с помощью простой панели инструментов. Одна из самых изящных функций – панорамный режим, который позволяет соединять фотографии, используя автоматическое совмещение и изображения, и цвета. Если, например, у вас есть две фотографии с пейзажем, отличающиеся яркостью и углом съемки, *Fotox* проделает достойную похвалы работу по соединению этих фото.

Самозащита

Firefox, возможно, безопаснее, чем другие браузеры, однако мы не должны спускать глаз с проблем, связанных с безопасностью. Mozilla Foundation недавно выпустила обновления для своих программ, и их можно найти на нашем DVD в разделе **Интернет**; мы настоятельно рекомендуем вам установить их, если ваш дистрибутив еще не выпустил пакетов обновления. При обновлении программ из внешнего источника, вроде LXFDVD, очень важно позаботиться, чтобы новая версия не перепуталась со старой, поэтому здесь мы покажем вам, как обновить *Firefox* и *SeaMonkey* с минимумом трудов.

Для *Firefox*, откройте менеджер пакетов вашего дистрибутива и полностью удалите браузер – частенько он сидит в **/usr/lib/firefox**, так что стоит проверить, существует ли эта директория после того, как вы удалили пакеты.

Теперь возьмите **firefox-2.0.0.5.tar.gz** из раздела **Интернет/Firefox** нашего DVD, скопируйте в домашнюю директорию, откройте терминал и перейдите в режим root с помощью *su* или *sudo bash*. Затем введите

```
cd /usr/lib
tar xfvz /home/имяпользователя/firefox-2.0.0.x.tar.gz
ln -s firefox/firefox /usr/bin/firefox
```

Объясняем: это переключит вас в директорию **/usr/lib** и затем извлечет туда новую папку *Firefox* – не забудьте сменить «имяпользователя» во второй строке, чтобы оно соответствовало вашей учетной записи. Третья строка создает ссылку в каталоге исполняемых программ (**/usr/bin**), чтобы вы могли запускать браузер, просто набрав '*Firefox*' (вместо того, чтобы вводить весь путь).

Можете применить ту же процедуру к пакету *SeaMonkey* – естественно, изменяя соответственно имена файлов там, где это необходимо. На нашем диске вы найдете релиз 1.1.2, который, кстати, стоит испробовать, если вы никогда не использовали его раньше. Как и оригинальный пакет *Mozilla*, он обеспечивает и браузер, и почтовый клиент, и стал сейчас так же быстр, как *Firefox* и *Thunderbird*.



› Добавьте чуточку жизни (морской или какой-нибудь еще) на своей рабочий стол с помощью *Wallpapoz*, и любуйтесь новым фоном каждые пять минут!

И напоследок...



Наконец, вас ожидает подборка лучших новых игр. Если вы интересуетесь старинными тренажерами полетов в стиле *Top Gun*, советуем вам *Thunder & Lightning*, впечатляющий самолетный имитатор, победивший три измерения. Скопируйте файл **TnL-linux-070526.tar.gz** в домашнюю директорию, распакуйте его и запустите tnl в директории **TnL/bin**. Подробное руководство – на http://tnlgame.net/content/view/21/53.

Для тех, кому по душе более традиционные платформенные развлечения, советуем *BlobWars*, не лишенную юмора платформенную игру, где более 25 миссий. В качестве Blob-агента Боба вы бродите по вражеским базам, сражаясь с другими Blob-существами-пузырями – чем не важное занятие! Хотите чего-то поумнее, попробуйте *Brikx*, развлекательную головоломку со своеобразной стилистикой. LXF



› О-о, эти эффекты пламени, как же истребителю без них!

Информация о диске

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ DVD-ДИСК !

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

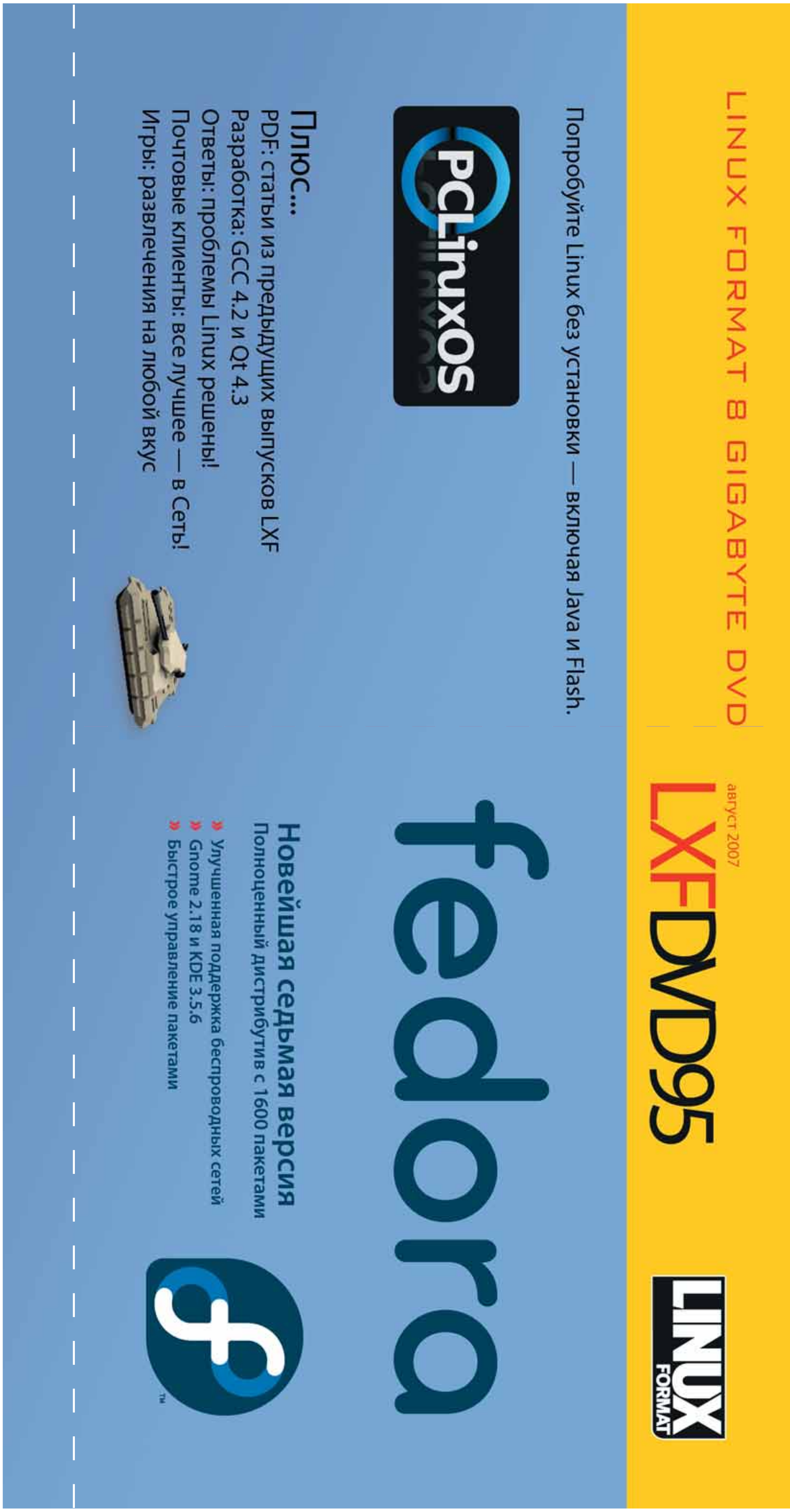
имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



DVD

Создание установочных дисков при помощи CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

Содержание LINUX FORMAT DVD Август 2007

LINUX FORMAT Июль 2007

Сторона 1

Аудио
Christine - аудиоплеер и каталогизатор музыки
GSSMP - аудиоплеер для среды GNOME
SoundKonverter - KDE-интерфейс для различных звуковых конвертеров
Traverso - мультитрековый аудиоредактор

Рабочий стол
F-Spot - каталогизатор фотографий
Gnome_Specimen - программа для отображения шрифтов
Gpredict - инструмент для слежения за спутниками
Gscan2pdf - утилита для преобразования отсканированных документов в PDF
TweaK - редактор скрытых настроек KDE
Wallpapoz - динамическая смена обоев для GNOME

Дистрибутивы
Fedora 7 - новая версия свободного дистрибутива от Red Hat
PCLinuxOS - популярный на западе десктопный дистрибутив

Разработка
Accerciser - интерактивный проводник по интерфейсным элементам GNOME-приложений
Edyuk - среда разработки приложений
GCC - коллекция компиляторов GNU
Konch - утилита для создания апплетов области уведомления
Mono - открытая реализация платформы .NET
MonoDevelop - свободная среда разработки для Mono
Nemiver - графический интерфейс к gdb
Qt - инструментарий для создания кроссплатформенных приложений

Игры
AstroMenace - космический шутер с отличной графикой и спецэффектами
BlobWars - кровавый шутер
Brikx - головоломка
JGames - сборник аркадных игр на Java
Snowballz - стратегическая игра
Thunder&Lightning - футуристический трехмерный авиасимулятор

Графика
Fotox - программа для улучшения изображений, снятых цифровой камерой
qLabels - программа для создания визиток
Veusz - научный пакет для создания графиков

Справка
Ответы на часто задаваемые вопросы
RUTE - руководство по администрированию Linux

Hot_Picks
Clonezilla - инструмент для клонирования системы
Eric - среда для разработки программ на языках Ruby и Python
GraphMonkey - калькулятор с поддержкой рисования графиков
KSquirrel - просмотрщик изображений для KDE
Misfit_Modeler - редактор 3D-моделей с поддержкой OpenGL
PeaZip - кроссплатформенный архиватор
Sketsa - кроссплатформенный векторный редактор
Sturmbahnfahrer - автосимулятор
TaxiPilot - симулятор таксиста
TorK - центр управления Tor для среды KDE

Интернет
Empathy - библиотека и Jabber-клиент
Firefox - популярный веб-браузер
Mldonkey - кроссплатформенный клиент для множества p2p-сетей
News_Notification - уведомление о новостях в RSS
Sancho - графический интерфейс для MLDonkey
SeaMonkey - набор Интернет-приложений
Sunbird - календарь-органайзер

Система
Avast-ng - фильтр логов
KNemo - сетевой монитор для среды KDE
RPM_Package_Maker - интерфейс для быстрого создания RPM-пакетов
SBackup - простое приложение для создания резервных копий
Wine - открытая реализация Windows API

Сторона 2

Slackware 12 - новая версия самого консервативного дистрибутива Linux
Zenwalk 4.6.1 - основанный на Slackware десктопный дистрибутив
ZenServer 0.5 - серверная версия Zenwalk GNU/Linux

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен на Уральском электронном заводе., 620066, Россия, г. Екатеринбург, ул. Комвузовская 17-203, Лицензия МПТР России ВАФ № 77-15

Поставляется вместе с журналом LINUXFORMAT номер 8(95) Август 2007

Содержимое DVD навскидку

ЖУРНАЛ
Akelos — Предварительный релиз.
Gtk — Примеры из Учебника GTK+.
LXFSchool — Материалы Школы LXF.
Mono — Код примеров статьи.
Java EE — DOJO, DWR, GWT и примеры их использования.
PDFs — Статьи из предыдущих номеров LXF.
Roundup — Email-клиенты.

ЗВУК
Christine — Медиа-плейер на основе GStreamer.
GSSMP — Скромный музыкальный плейер.
SoundKonverter — Интерфейс для аудиоконверторов.
Traverso — Станция по работе с цифровым звуком.

РАБОЧИЙ СТОЛ
F-Spot — Персональный инструмент управления фотографиями.
Gnome Specimen — Программа предварительного просмотра шрифтов.
Gpredict — Отслеживание спутника и предсказание орбиты.
Gscan2pdf — Создание многостраничных PDF со сканера.
Tweak — Менеджер настроек KDE.
Wallpapoz — Утилита работы с обоями Gnome.

РАЗРАБОТКА
Accerciser — Accessibility-утилита.
Edyuk — Интегрированная среда разработки.
GCC — Коллекция компиляторов GNU.
Konch — Создатель апплетов системного лотка.
Mono — Среда .NET с открытым кодом.
MonoDevelop — Среда интегрированной разработки C#.
Nemiver — Автономный графический отладчик.
Qt — Среда разработки приложений.

ДИСТРИБУТИВЫ
Fedora — Дистрибутив от Red Hat.
PCLinuxOS — Live- и устанавливаемый дистрибутив.

ИГРЫ
AstroMenace — Космическая стрелялка.
BlobWars — Платформенная игра на основе SDL.
Brikx — Головоломка.
JGames — Компиляция игр в жанре action.
Snowballz — Стратегия в реальном времени.
Thunder & Lightning — Тренажер полетов.

ГРАФИКА
Fotox — Улучшение цифровых фотографий.
QLabels — Программа создания ярлыков и визиток.
Veusz — Пакет для построения научных графиков.

СПРАВКА
Answers — Решенные проблемы Linux.
Rute — Руководство по администрированию Linux.

HOTPICKS
Clonezilla — Программы клонирования диска.
Eric — Интегрированная среда разработки Python и Ruby.
GraphMonkey — Калькулятор графиков и схем.
KSquirrel — Программа просмотра изображений для KDE.
Misfit Modeler — Редактор трехмерных моделей на базе OpenGL.
PeaZip — Менеджер архивов.
Sketsa — Редактор векторной графики.



› Обзаведитесь собственной студией звукозаписи с помощью Traverso (в разделе Звук).

Sturmbahnfahrer — Автомобильный тренажер.
TaxiPilot — Игра в жанре action.
Tork — Менеджер анонимности.

ИНТЕРНЕТ
Empathy — IM/голосовой/видео клиент Gnome.
Firefox — Web-браузер.
MLDonkey — P2P-клиент.
News Notification — Новости RSS во всплывающем окне.
Sancho — GUI-интерфейс для MLDonkey.
SeaMonkey — Web-браузер и email-клиент.
Sunbird — Календарь Mozilla Project.

СИСТЕМА
Avast-ng — Программа просмотра системного журнала с фильтрами.
KNemo — Сетевой монитор KDE.
RPM Package Maker. — Утилита создания пакетов RPM.
Simple Backup — Утилита создания резервных копий файлов.
Suite Wine — Программа запуска программ Windows в Linux.

ГЛАВНОЕ
ATI driver — Графический драйвер.
Bash — Оболочка командной строки.
Cairo — Библиотека двухмерной графики.
CheckInstall — Программа создания двоичных пакетов.
Coreutils — Утилиты командной строки.
GLib — Низкоуровневая библиотека.
GTK — Инструментарий пользовательского интерфейса.
HardInfo — Системная информация и сравнительный анализ.
Jigdo — Создатель ISO-образов.
Kernel — Свежий релиз ядра Linux.
LibXML2 — XML-анализатор и инструментарий.
Ncurses — Оконный инструментарий текстового режима.
Nvidia driver — Графический драйвер.
Rawrite — Программа записи образов на дискеты.
SBM — Smart Boot Manager.
SDL — Библиотека мультимедиа.



› Blob Wars: войдя в роль бесстрашного агента Blob (Пузыря), задайте жару Пузырю-врагу.

Школа LINUX FORMAT

Пожарина Галина Юрьевна,
начальник отдела информационных технологий,
Санкт-Петербургский технический колледж
управления и коммерции

Переводим образование на свободное ПО?!

Сектор свободного программного обеспечения интересен и важен для образовательной сферы по нескольким причинам. И решение вопроса информационно-правового порядка при оптимальных финансовых затратах – отнюдь не не единственное его преимущество.

Свободное программное обеспечение находится в состоянии постоянного совершенствования. Программы оперативно исправляются и дополняются, при этом новые релизы по условиям лицензии обязательно публикуются, так что имеется реальная возможность улучшить качество программного продукта в короткие сроки.

Используя любую программу из сектора свободного ПО, преподаватели и учащиеся исследуют современное, актуальное решение. Установка заведомо «старых» программ (таких как Microsoft Office 2003 – уже четырехлетней давности) провоцирует тормозящий путь обучения – оно ведется пусть и на устойчивой версии программного продукта, пусть и с богатым методическим обеспечением, но, поскольку по окончании учебы пользователь (бывший школьник или студент) будет использовать уже более свежие релизы, ситуация спровоцирует необходимость дополнительного повышения квалификации. Это нивелирует ценность основного базового образования.



С позиции технического сопровождения сложностей в работе со свободным программным обеспечением не возникает, поскольку обновление программ производится зачастую через Интернет в автоматическом режиме, что не сказывается на основной деятельности пользователя: нет необходимости переустанавливать программу, не теряются пользовательские настройки и сервисы.

Привлекательность свободного программного обеспечения для образовательных целей заключается еще и в том, что, поскольку авторами большого количества программных продуктов выступают учащиеся (студенты и студенческие коллективы), то и выполненные в качестве курсовых и дипломных проектов свободные программы создаются в качестве решений практически значимых для обучения задач и отвечают потребностям и запросам образовательного процесса.

Учитывая тот факт, что работы оцениваются как учебные, к ним применим и критерий соответствия образовательному стандарту. Если программный продукт имеет прикладной характер, то учтены все необходимые и достаточные функции, способные обеспечить изучение конкретной темы, будь то таблица Менделеева, конструктор молекул или клавиатурный тренажер.

Интеграция в обучение программного обеспечения связана для методиста и преподавателя со следующими действиями:

- изучение литературы по проблеме;
- поиск и подбор программного обеспечения;
- установка программы и адаптация к условиям конкретного компьютерного класса или рабочего места преподавателя;
- поиск или составление методических материалов.

К сожалению, ресурсы глобальной сети в данной сфере достаточно разрозненны.

Чтобы понять, что преподаватель не окажется один на один с техническими проблемами и вопросами, на которые непросто сразу найти ответ, зайдите в сетевой журнал Владимира Иванова (**http://ivlad.livejournal.com/**) – кстати, вы могли встретить его в рубрике Ответы LXF. Автор не только рекомендует несколько интересных ресурсов, но и инициировал сообщество специалистов в области Linux и свободного программного обеспечения, готовых консультировать учителей и руководителей школ не

только в Москве и Санкт-Петербурге. География участников сообщества достаточно обширна.

Федерация интернет-образования анонсировала проект СПООР (Свободно распространяемое программное обеспечение для образования России). Адрес проекта: **http://uchim.info/spoor/index.htm** Ресурс содержит статьи, пресс-релизы и методические материалы для педагогов и руководителей образовательных учреждений.

По адресу **http://edu.of.ru/profil/default.asp?ob_no=18500** расположен сайт программы «Интернет-поддержка профессионального развития педагогов» РГПУ имени А.И. Герцена. Формирующаяся на страницах этого проекта подборка материалов и ссылок по тематике свободного программного обеспечения адресована именно учителям и методистам.

«Инфра-ресурс» (**http://www.i-rs.ru**) публикует материалы, связанные с внедрением пакета OpenOffice.org. Проведенный по инициативе неформального сообщества разработчиков и пользователей Ru.OpenOffice.org при поддержке журналов Linux Format и «Мир ПК» конкурс статей и методических пособий о пакете OpenOffice.org позволил собрать и опубликовать на сайте материалы, связанные с использованием этого офисного пакета.

Системы компьютерной алгебры (такие как Mathematica, Maple, Axiom и MAXIMA и др.) уже давно перестали быть игрушками для студентов, а стали серьезным рабочим инструментом для аналитиков. Сообщество MAXIMA (**http://community.livejournal.com/maxima_platform**) приглашает принять участие в обсуждении одноименной свободной системы символьных вычислений(см. LXF81-86).

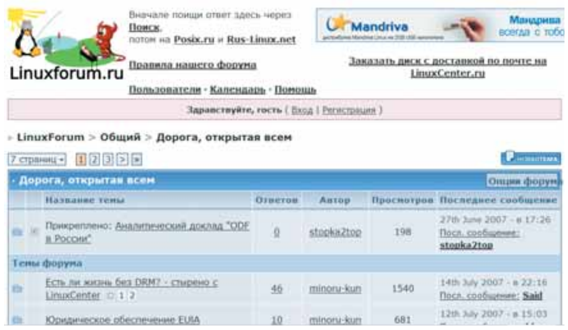
Волгоградский Linux User Group делится опытом внедрения в регионе. Адрес проекта: **http://volgograd.lug.ru/wiki/MicrosoftCaseStudyComments**

Для тех, кто хочет познакомиться с обширным аналитическим материалом, мы рекомендуем сетевые библиотеки CITKIT (**http://citkit.ru**), проектов LinuxRSP (**http://www.linuxrsp.ru/**) и Линуксцентра (**www.linuxcenter.ru**)

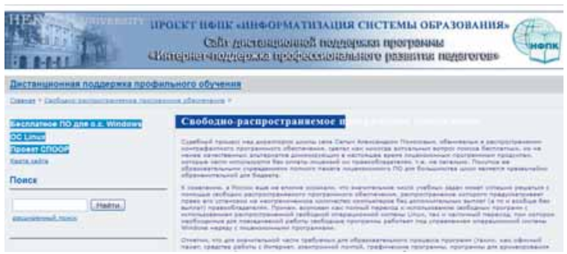
Живое общение и консультации профессионалов на форуме **http://www.linuxforum.ru**

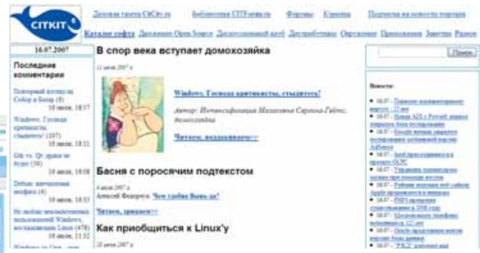
Приведенный перечень ресурсов далеко не полон. Но и этот краткий обзор дает возможность запустить проекты по внедрению свободного программного обеспечения в ваши школы, лицеи, техникумы, колледжи, вузы. Пора начинать.







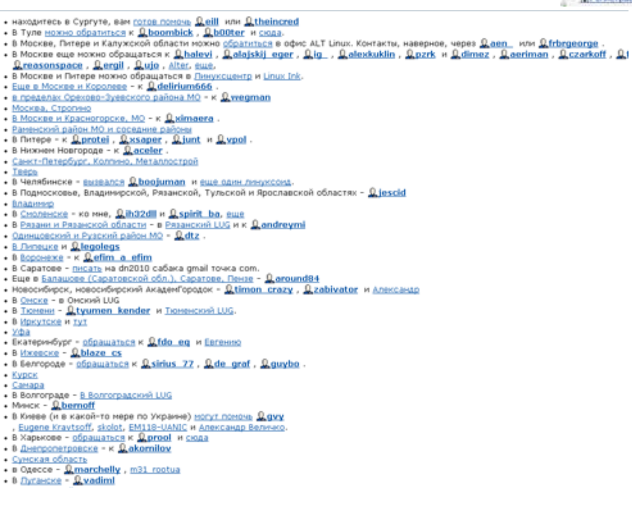


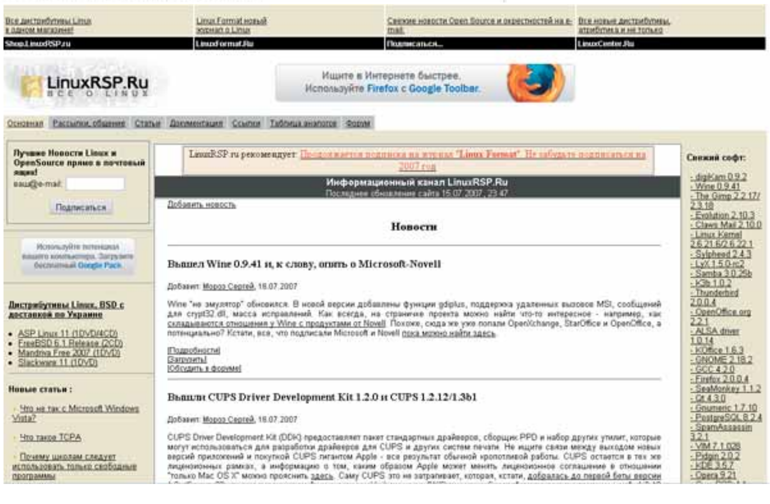












АЛЕКСАНДР МИХАЙЛОВИЧ ПОНОСОВ

«У САМОВАРА»



Получив предложение вести авторскую рубрику в Linux Format, я крепко призадумался: о чем может поведать читателям новичок не только в мире Open Source, но и в журналистике. Обратился за советом к друзьям, реальным и виртуальным, для примера приведу несколько пожеланий из сетевого дневника:

«Надо попытаться шагнуть на встречу обычному пользователю, чайнику. Хотя бы делать обзоры по Open Source в сфере образования.»

«с тематикой как раз понятно – как n00b становится специалистом.»
«Я так понял, Вы сейчас внедряете у себя Linux в школе. Вот и пишите, с какими трудностями сталкиваетесь. Не только техническими, но и юридическими, и прочими.»

«Александр, во-первых, подойти творчески.»

«...как преподавать информатику с использованием Linux. Вот это было бы полезно и интересно».

На ближайшую перспективу я решил остановиться на следующем варианте. Центральное место в моих публикациях займут вполне конкретные люди самых разных профессий, так или иначе связанные с продвижением в жизнь свободного программного обеспечения.
Исходя из общепринятого определения «чайник», коим я себя по праву могу считать, рубрику назовем условно «У самовара».
Что планируется? Обсуждение «за чашкой чая» различных вопросов по использованию продуктов Open Source в образовании, исследование проблем, с которыми сталкиваются «первопроходцы», анализ имеющегося опыта миграции на СПО и, наконец, просто «живое» общение с интересными людьми.

Для первого раза разрешите немного о себе.

Как я стал адептом Open Source

или

История миграции на Linux в одной отдельно взятой голове

Мои ИТ-университеты

Первое знакомство с ЭВМ состоялось в далеком 1983 году. Уральский политехнический институт. «Основы вычислительной техники». Фортран. Алгол. Блок-схемы. Мало чего помню, но вроде как получалось и нравилось. Кафедра тогда была не выпускающая, а то, возможно, я бы не оставил учебу на химфаке, а просто перевелся на другой факультет.

Через 12 лет директор Сепычевской школы отправила меня на курсы техников-лаборантов кабинета информатики. Не судьба: учебу отменили, не набрали группу. Тем не менее, успел «пощупать» IBM, их тогда ставили головными машинами на класс «Электроника»,

Историко-политологический факультет ПермГУ. 1997 год. Зачет по информатике сдавал как типичный студент-двоечник. В тоже время однокурсники со школьной скамьи чувствовали себя в компьютерном классе как рыба в воде. Как говорится, «зацепило». Поэтому, как только в конце 1999 года на смену «Электронике» в школу поступил комплект современных машин, обложился самоучителями и пустился, «задрав штаны, бежать за комсомолом». К 2004 году, когда меня назначили директором, для решения своих задач с компьютером я общался уже более уверенно. Что касается вообще информационных технологий, то по сути остался самоучкой: «тут знаю, там не знаю, но, понадобится, тоже узнаю».

По жизни придерживаюсь принципа: «Если нечто придумал, человек, другой человек в состоянии с этим разобраться».

«И грянул гром...»

В 2005 году пустили в эксплуатацию I очередь новой школы, которую строили почти 30 лет.

В числе оборудования – комплект компьютеров. Программы предустановлены, никакого подозрения на их нелегальность. Мои же познания в области авторского права в то были на уровне обывателя: ловят и наказывают распространителей «пиратских» CD, DVD, всякого рода плагиаторов. Прокурорская проверка стала шоком, начинка компьютеров была объявлена контрафактной со всеми вытекающими последствиями. Думаю, что подробности «дела Поносова» повторять нет смысла, остановлюсь на некоторых деталях, которые в конечном итоге привели меня к Linux и Open Source.

Шахматная вилка для «виндузятника»

Работники прокуратуры изымать до августа компьютеры не стали, тем не менее на всякий случай взяли с меня расписку следующего содержания: « МОУ «Сепычевская средняя общеобразовательная школа» обязуется не использовать и не удалять нелицензионные программы на компьютерах...» Как показали дальнейшие события, провокация чистейшей воды, благо, что в школе компьютеров в два раза больше, чем в уголовном деле, да хватило выдержки не уничтожать улики.

Осенью после экспертизы жесткие диски в качестве вещественных доказательств вернули в школу. Головоломка осталась: как не сорвать учебный процесс, при этом соблюсти законность . Решения, понятно, были шаблонные. На деньги со статьи «наглядные пособия» закупил в рассрочку комплект лицензионных программ Microsoft. На что устанавливать? Установка – форматирование – уничтожение вещдоков – статья УК РФ. «Выревел» в долг новые жесткие диски. Одним словом, выкрутился. Но задачка в голове осталась: как быть в ладу с законом, если нет возможности (времени) приобрести легальные программы, нет других HDD, и нельзя прекращать использование компьютеров? Можете смеяться, но до декабря 2006 года я не знал о какой-либо альтернативе ПО Microsoft.

До конца первой серии судебных баталий я не допускал резких выпадов в сторону Microsoft, следуя логике Льва Троцкого: «Войны не ведем, мира не подписываем, армию не демобилизуем.» Вообще, на этом этапе можно было умыть руки: проблема актуализирована, осталось покаяться в грехах и удалиться сеять разумное, доброе, вечное. Но...»Поносова понесло».

Когда стало ясно, что корпорации все равно нужен мальчик для битья, открыто заявил о переходе в лагерь FOSS, за что г-н Елашкин намекнул на мое сходство с Полиграф Полиграфычем. Может быть, со стороны все происходящее выглядело по Булгакову: «линуксоиды» вылепили из «обиженного» директора идейного борца с заокеанским монополистом. Только Швондером на этот раз был я сам. Залп критики типа «Вы ИТ-профессионал? Так сидите и не п...те!» лишь убедил меня плотнее заняться самообразованием.

Знакомство с Linux: «Так вот ты какой, северный олень!»

Первый раз я увидел Linux «вживую» на ноутбуке Даниила Курушина, программиста из ПермГТУ. Он в по просьбе Министерства образования края консультировал моего адвоката. Кроме вопросов, непосредственно касающихся судебного разбирательства, мы подробно обсудили возможность использования свободного программного обеспечения в школах в различных вариантах. Затем из Санкт-Петербурга прислали первый дистрибутив Ubuntu, его я за полным отсутствием опыта попробовал в режиме Live CD. Серьезно занялся изучением системы, когда получил от компании Linux Ink коробку Scientific Linux с руководством и несколько номеров Linux Format с DVD. Поэкспериментировав с разными дистрибутивами, в конечном итоге оставил Scientific, наверное, по принципу «лучший дистрибутив – это тот, который хорошо знает ваш гуру», при решении сложных вопросов можно было рассчитывать на помощь Linux Ink.

Кстати, установка SL на пять школьных компьютеров прошла без заморочек. Для того, чтобы не пугать народ, разместил его в параллель с Windows, правда особого интереса пока не наблюдаю. Проблемы с «железом» имеются только на ноутбуке: не работает встроенный модем, какой-то глюк с видеокартой, работаю с маленьким разрешением экрана. Трагедии из этого не делаю, дойдут руки, займусь.

> **«До декабря 2006 года я не знал о какой-либо альтернативе программам Microsoft.»**

А король-то голый!

Когда стало ясно, что что система образования защищать меня явно не собирается, корпорация на компромисс идти тоже не желает, сел за директорский ноутбук и испустил виртуальный вопль.

24 декабря 2006 года. Общероссийский образовательный портал.

«...А как дела с лицензиями по России, неужели только у нас «паленые» программы?»

Обсуждение этой темы показало, к сожалению, что в России появилась третья беда. Как всегда на повестке дня встали два вопроса: « Кто виноват?» и »Что делать?»

А мальчика из сказки скорее всего показательно выпороли...

«Назло бабушке отморожу уши!»

Прочитав на форумах первые сообщения о возможности перехода на свободное программное обеспечение, несмотря на катастрофическую нехватку времени, стал изучать все доступные ресурсы по Open Source. Отправной точкой стала статья Ричарда Столлмэна «Почему школы должны использовать исключительно свободное программное обеспечение «.

Так уж получилось, что сначала я сначала проникся идеологией СПО, разобрался с терминологией и различными типами лицензий и уже после этого перешел к практике. Тем не менее внутренний выбор был сделан.

Я NOOB, но это небезнадежно.

Мое заявление «за месяц освоил Linux», растиражированное СМИ, само собой, не стоит понимать буквально, для уточнения я бы добавил «освоил снаружи». За прошедшие полгода я преодолел главное – психологический барьер, неизбежный для первого шага. Все реже возвращаюсь к Windows и MS Office, все чаще почитываю умные книжки, на очереди изучение Linux «изнутри». Решение в пользу Open Source, принятое не без влияния эмоций, остается в силе.

Linux в школе

В общеобразовательной школе поселка «Горки-10» вот уже шесть лет работает «семейный подряд»: мать – учитель информатики и сын – системный администратор. **Евгений Балдин** решил разобраться, почему Linux сделал их жизнь проще.



Как обустроить класс

Система GNU/Linux в школе хороша – и отнюдь не только потому, что бесплатна. GNU/Linux – идеальный помощник для учителя в его желании обучать.

Для продвижения GNU/Linux в школе следует привлекать не столько детей, сколько преподавателя. Необходимо постоянно подчеркивать сильные стороны этого образчика программного обеспечения: он легко настраивается под задачу, он работает на исправных компьютерах почти независимо от их года выпуска, а при грамотной установке поддержка занимает удивительно немного времени.

Что дано:

В кабинете информатики средней общеобразовательной школы «Горки-10», расположенной в одноименном обычном поселке на двадцать пятом километре необычного Рублево-Успенского шоссе имеется свой компьютерный «зоопарк»:

- Celeron 2000 МГц / 256 МБ ОЗУ / NVIDIA MX200 32МБ / 60 ГБ HDD – 3 компьютера,
- Celeron 733 МГц / 128 МБ ОЗУ / Trident Blade 8 МБ (встр.) / 20 ГБ HDD – 5 компьютеров,
- Pentium III 850 МГц / 128 МБ ОЗУ / NVIDIA Vanta 16 МБ / 30 ГБ HDD– 1 компьютер,
- Pentium III 733 МГц / 128 МБ ОЗУ / NVIDIA Riva 32 МБ / 40 ГБ HDD – 1 компьютер,
- Pentium II 266 МГц / 192 МБ / NVIDIA Riva 32 МБ / 5 ГБ HDD – 3 компьютера,
- 8-портовый хаб на 10 Мбит,
- 6-портовый свитч на 100 Мбит.

При этом есть проблемы с сетью, так как хаб и разъемы RJ45 были заляпаны водоэмульсионкой во время ремонта, после чего в целях очистки рабочие «вымыли» все это водой. Выхода в Интернет нет.

На всем этом учитель должен наладить образовательный процесс в соответствии со стандартом. Стандарт, возможно, «кривой», но все же стандарт.

Реальное решение:

В качестве основного дистрибутива был выбран основанный на *Slackware* компактный *VectorLinux 5.0* (http://www.vectorlinux.com/). В качестве источника пакетов для основного дистрибутива служит *DeepStyle 1.0* (http://deepstyle.org.ua/) – локализованный и немного доработанный украинскими энтузиастами *Slackware*.

Вход в систему осуществляется через *xdm*. На каждом компьютере существуют два пользователя: guest и guest7. Учетная запись выбирается в зависимости от темы урока. Домашние каталоги запакованы в tar-архивы и распаковываются при каждой перезагрузке X-Window. Для этого в файл **Xsetup_0**[1] добавлен вызов скрипта **clear.sh**:

```
#!/bin/sh
/usr/bin/rm -rf /home/guest
/usr/bin/tar -xzf /home/guest.tgz -C /home
/usr/bin/rm -rf /home/guest7
/usr/bin/tar -xzf /home/guest7.tgz -C /home
```

Эти простые настройки гарантируют неизменность состояния рабочего места ученика на начало урока, так как переписывается абсолютно все.





Первое, что видит ученик после авторизации – это рабочий стол:

- Легкий оконный менеджер *IceWM* (http://www.icewm.org/) с темой Xp-Silver.
- Никаких иконок.
- Неизменяемое меню, в котором содержатся только названия программ и пункт «Выход».

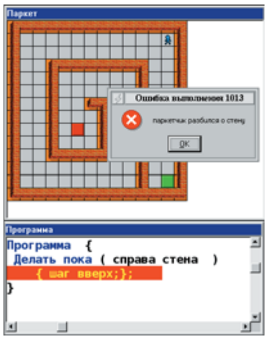
[1] Примеры настройки можно посмотреть в **man xdm**.

— это удобно

> **LINUX FORMAT На DVD**
> » План занятий по информатике (7-11 кл.) от Чемисовой Г.Н.

» Выключены виртуальные рабочие столы и все значки на панели быстрого запуска.

В меню учащийся при желании или по команде учителя может отыскать следующие программы:



» ***Паркетчик*** – это рекомендованный учебный язык программирования, доступный по адресу http://www.inf777.narod.ru/parketchik.htm. Запускается под *Wine* (LXF94).

» ***Аленка*** – очень простой клавиатурный тренажер, созданный около 17 лет назад. Запускается в *DOSBox* (LXF94).



» ***KTouch*** – сложный клавиатурный тренажер. Русская раскладка и уроки входят в стандартный дистрибутив. Является частью KDE.

» ***SmallBasic*** – качественная кроссплатформенная реализация учебного языка Basic (http://smallbasic.sourceforge.net/).

» ***Gambas*** – интерпретатор языка Basic, интегрированный в среду разработки, с возможностью создания графического интерфейса (http://gambas.sourceforge.net/).

» ***Gkrellm*** – монитор нагрузки системы. Имитация программы, отображающей нагрузку в Windows.

» ***Gcalctool*** – калькулятор. Пример всплывающих подсказок и выяснение по ним неизвестных функций. Является частью GNOME.

» ***Stardict*** – словарь (http://stardict.sourceforge.net/ – словари там же). Используется для обучения работы с электронным словарем.

» ***XFE*** – простой файловый менеджер (http://roland65.free.fr/xfe/). Имитация программы *Explorer* и *Norton Commander*.

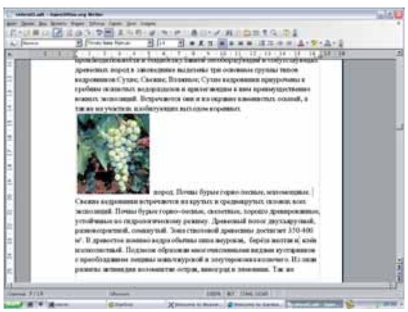
» ***Kolourpaint*** – программа рисования. Рисование основных примитивов, масштабирование, искажения и заливка цветом. Имитация программы Paint. Является частью KDE.

» ***Firefox 1.5*** с подключаемыми модулями Java и Flash. Так как доступа в Интернет нет, на учительском компьютере установлен *Apache*. Для генерации примеров сайтов и обучения навыку поиска используется *CMSsimple* – несложная система управления контентом (http://www.cmsimple.dk/).

» ***Beaver*** – простой HTML-редактор (http://www.nongnu.org/beaver/). Есть подсветка синтаксиса.

» ***OpenOffice.org 2.0*** – офисный пакет, замена Microsoft Office.

» ***KBlackbox*** – логическая игра, пример черного ящика. Является частью KDE.



Пакеты устанавливаются в полуавтоматическом режиме с ftp-сервера учителя. Решение с ftp-сервером оказалось самым простым, но ничему не противоречит, чтобы для этой же цели использовался *NFS*.

Пакеты подготавливаются с помощью makepkg. Для их установки создан простой скрипт **sc_installer**:

```
# !/bin/sh
#
# install all packets from ftp 192.168.0.1/pub/packets_to_istall
#
rm /tmp/inst_pack/*
mkdir /tmp/inst_pack
cd /tmp/inst_pack/
/usr/bin/wget ftp://192.168.0.1/pub/packets_to_istall/*.tgz
/sbin/installpkg *.tgz
rm /tmp/inst_pack/*
rmdir /tmp/inst_pack/
```

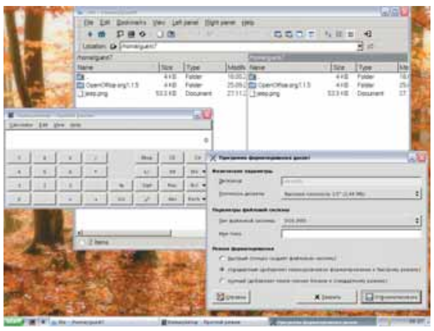
Не менее простой скрипт **sc_home** удаляет предыдущие архивы с настройками и копирует их с компьютера учителя:

```
# !/bin/sh
#
# install all packets from ftp 192.168.0.1/pub/new_home
#
cd /home/
rm *.tgz
/usr/bin/wget ftp://192.168.0.1/pub/new_home/*.tgz
```

Теперь для обновления и настройки программ достаточно выполнить:

```
> ssh root@user1
> sc_installer&&sc_home
```

Этот конструктор прост, надежен и главное – работает. Да, еще важно, чтобы права на файлы конфигурации были только у root, а у пользователя был доступ только на чтение.



Методическое обеспечение:

Предмет «Информатика» сейчас проходят с седьмого по одиннадцатый классы. Для обучения необходимы рекомендованные учебники. Выбор учебников остается, естественно, за учителем, но здесь предлагается уже опробованное решение:

» **7 класс** Н.Д. Угринович «Учебник для 7 класса общеобразовательных учреждений» (ISBN 978-5-94774-673-0).

» **8 класс** Н.Д. Угринович «Информатика и ИКТ: Базовый курс: Учебник для 8 класса» (ISBN 5-94774-630-1).

»

» **›› 9 класс** Н.Д. Угринович «Информатика и ИКТ: Базовый курс: Учебник для 9 класса общеобразовательных школ» (ISBN 978-5-94774-672-3).
›› 10–11 классы А.Г. Гейн, А.И. Сенокосов, Н.А. Юнерман «Информатика. 10-11 класс» (ISBN 5-09-011707-1/5-09-012178-8/5-09-014401-X).

Утверждается, что в учебнике Гейна, Сенокосова и Юнермана нет жесткой привязки к операционной системе. В учебниках Угриновича слово Linux упоминается прямо в аннотации, но к самому учебнику есть претензии в методическом плане.

К сожалению, учебник Гейна, Сенокосова и Юнермана выпал из «Перечня рекомендованных (допущенных) учебников и учебно-методических изданий» на 2006/2007 года.

Планы занятий, расписанные по урокам, можно посмотреть на прилагаемом к журналу диске. Естественно, их необходимо модифицировать под свои нужды. Достоинство данных планов в том, что они реально опробованы. LXF

Интервью: Галина Николаевна Чемисова

ГАЛИНА НИКОЛАЕВНА ЧЕМИСОВА:
«Всему миру помочь невозможно, поэтому начинать улучшать его следует со своих близких. Делай то, что ты умеешь лучше окружающих – это повышает ценность хороших поступков.»

Евгений М. Балдин: Вы можете как-то сравнить ситуацию «до» и «после»?
Галина Н. Чемисова: Ситуация сложилась так, что большое количество лет я преподавала в классе УК-НЦ под ФОДОСом[4]. Первоначально в новый класс поставили 5+1 компьютер и хаб на 8 портов с предустановленной Windows 2000.

Проблема заставить детей делать только то, что требуется, решалась созданием гостевого профиля с удалением из главного меню и с рабочего стола всего ненужного. После перезагрузки восстанавливались установки из «Default user». С самого начала подготовки и настройки программного обеспечения мне помогал сын, он же решал все проблемы с «железом» (подключить, раскрутить, проверить сборку, заменить сетевые карты, поменять сетевой кабель и т.п.).

Пока ученических ПК было пять, проблем, в общем, не было, потому что большая часть занятий проходила в классе УКНЦ. Проблемы начались, когда в класс начали ставить (в разное время) разное количество компьютеров. На настоящий момент в классе есть шесть разновидностей конфигураций системных блоков и семь разновидностей мониторов, хаб 10-мегабитный на 8 портов и 100-мегабитный свитч на 8 портов.

Проблема еще состояла в том, что мы никак не могли настроить работу сети под Windows. Количество часов, которое мы на это угробили, не поддается никакому подсчету. Кроме того, после УКНЦ очень раздражало то, что детей невозможно заставить заниматься делом[5]. Отключать правую кнопку мыши, как это делают многие «озверевшие» учителя я считала нецелесообразным, так как не вырабатывается важный навык работы с контекстным меню. Самые бестолковые дети, которые не могли найти игрушки, но не желали работать по теме урока, рисовали в *Paint* всякую ерунду и делали ее фоном рабочего стола. Некоторые переименовывали папки на рабочем столе, настраивали хранителя экрана – бегущую строку со всяким непотребством.

С самого начала в качестве второй операционной системы был установлен Linux (на первых 6 компьютерах). Пробовались разные дистрибутивы: Alt Linux Junior, Mandrake. Последней каплей, вызвавшей срочный переход на Linux, стала установка трех P2-266 с 5-ГБ дисками и проблемами с оборудованием (не могли найти драйверы к сетевой карте). Ударными темпами в сентябре перешли на Vector Linux и *OOo 1.5*.

Евгений: Как это повлияло на проведение уроков? Стало ли легче?
Галина: Поскольку учебных часов на каждый класс мало (стандартно 1 час в неделю и только один класс 2 часа в неделю, но у них информатика началась в 10-ом), у меня трудностей не было. Многие дети до сих пор не поняли, что они работают в другой операционной системе, некоторые дома перешли на *OpenOffice*.org (2 или 3 человека). Основная претензия к GNU/Linux у детей – их убеждение в том, что под Linux нет игрушек. Аргументы, что под Windows 2000 и XP первоначально тоже не было игрушек, не воспринимаются.

Мне же стало работать на порядок легче, так как из настроек базового пользователя удален даже растровый графический редактор, а в векторном дети, когда им нечего делать, почему-то не рисуют. Перестала дергаться, не сделал ли кто-нибудь из детей за время урока какую-нибудь гадость. Отпала необходимость постоянно обновлять антивирус, а это на 13 компьютерах при неработающей сети и разном оборудовании целая проблема. Правда осталась проблема: «особо одаренные», когда не могут выполнить задание в электронных таблицах, пытаются, выделяя диапазоны ячеек, писать какой-то текст, как правило, не очень литературный.

Евгений: Как отнеслась к нововведению администрация? Другие учителя? Повлияло ли на Вас история с Поносовым?
Галина: Никак. Администрация считает, что проблемы кабинета ее не касаются. Это я должна оказывать посильную помощь в освоении ПО, мои же проблемы никого не волнуют. Группа учителей в количестве 12 человек прошла 36-часовой курс начинающего пользователя. Один человек после этого перешел на Linux. История с Поносовым привела к обнулению win-разделов на всех жестких дисках, т.к. доказать, что мы этим не пользуемся, будет невозможно.

Евгений: Как относятся к текущей ситуации дети? Были ли сложности с переходом для них?
Галина: Им все равно.

Евгений: Заинтересовался ли кто-нибудь из детей системой, которая стоит на компьютере в классе? Попробовал ли кто-то ее дома?
Галина: Иногда задают вопросы: «Почему в классе все получается, а дома – нет?»

Евгений: Что бы Вы пожелали или посоветовали с высоты своего опыта?
Галина: Я советую всем перейти на открытое программное обеспечение, так как считаю неверным отдавать те крохи, которые выделяются на информатизацию в школе, за платное ПО. Кроме того, ни Windows 2000, ни Windows XP не позволяют спокойно жить учителю, так как периодически требуют переустановки, которая занимает до четырех часов личного времени на каждом компьютере. Ну и в довершение всего, эти операционные системы не позволяют проводить внятную сетевую политику.

В заключение хочу сказать, что я не работаю и не работала как системный администратор. Этим занимается сын. Мне как учителю стало работать проще.

---

[4] Клон RT-11 – операционной системы от фирмы DEC для PDP-11

[5] В классе УКНЦ все грузилось с учительского компьютера и существовали конфигурации под каждого ученика каждого класса и перед приходом нового класса нужно было просто смонтировать логический диск и, при необходимости, загрузить нужный урок на ученические компьютеры.

Интервью: Константин Александрович Чемисов



КОНСТАНТИН АЛЕКСАНДРОВИЧ ЧЕМИСОВ:
«Всему миру помочь невозможно, поэтому начинать улучшать его следует со своих близких. Делай то, что ты умеешь лучше окружающих – это повышает ценность хороших поступков.»

Чемисов Константин Александрович, инженер-технолог (машиностроение), внедрил GNU/Linux в учебный процесс в кабинете информатики школы Горки-10 более полутора лет назад.

Евгений М. Балдин: Как Вы оказались в школьном кабинете информатики?

Константин А. Чемисов: Преподавателем работает моя мама, поэтому приходится помогать. До установки нового класса работал класс УКНЦ[2] и с ним проблем не было. С новыми компьютерами проблемы появились сразу, и я вызвался помочь. Вот и помогаю уже около шести лет.



Евгений: Как и почему это было проделано? Какие проблемы встретились на этом непростом пути?

Константин: Началось все из-за желания «наличия отсутствия» проблем. Первым делом были написаны уроки для KTouch[3] (2 дня). Потом был выбран дистрибутив, и им оказался Vector 5.0.2 Standard. Первоначально сохранялась двойная загрузка, но после того как Linux был настроен, преподавание велось только под ним.

В процессе были проблемы с оборудованием (вздувались конденсаторы на материнских платах), но Linux их стойко пережил. *OpenOffice.org 1.5* был заменен на *OpenOffice.org 2.0*, когда тот вышел. Это было сделано из-за наличия в последнем базы данных. Были настроены «псевдо-сайты» на учительском компьютере и добавлены обои на рабочий стол.

Все было сделано так, что школьники не могут менять обои, не могут настраивать заставку, короче, ничего не могут, кроме положенного по теме урока. В сети используются статические IP-адреса и файл hosts. Последнее время думаю про DNS, но пока лень.

Евгений: Как переход на Linux облегчил работу учителя?

Константин: То, что ученик не может ничего испортить, сохраняет нервы преподавателя. Нет никаких идиотских фраз в хранителе экрана, никаких дурацких заставок из Paint. Поиграть им не дают. Есть только одна игра – KBlackbox, но и она запускается по SSH учителем, когда это требуется.

Евгений: Чем объясняется выбор основного дистрибутива?

Константин: Я начинал со *Slackware 3.5* и «слакварь» мне привычна. Всегда можно что-то «коряво отрубить», и потом все остальное будет работать. Vector Linux действительно быстрый и оптимизированный дистрибутив. К тому же стандартная версия представляет собой tar-архив с корневой системой внутри, которая довольно быстро распаковывается даже на медленных машинах (в отличие от дистрибутивов с кучей пакетов).

Евгений: Были ли возражения со стороны учеников? Фразы, типа «долго грузится» и тому подобное?

Константин: Сложности были, но только из-за непривычки. Хотя, *OpenOffice.org* действительно грузится долго и работает не спеша, особенно база данных. Возражения со стороны учеников были, но они смирились. Аргументы «против», как правило, на уровне: «Виндоус, потому что он нормальный».

Евгений: Родители не возмущались?

Константин: Родители вряд ли знают, на чем работают дети. Учителя даже проходили обучение на этих компьютерах и ничего, справились.

Евгений: Мне кажется реализованной вариант настройки довольно любопытен. Это где-то подсмотрели или собственная разработка?

Константин: Разработки все собственные (в них нет ничего сложного). Теперь рабочее место ученика представляет из себя стройную систему «костылей и подпорок» (улыбается).

Евгений: Что нужно GNU/Linux, для того чтобы продвинуться в школе?

Константин: Наверное, стандартный дистрибутив с автоматической установкой. Лучше на основе Debian.



[2] Электроника УКНЦ или УК-НЦ – учебный компьютер Научного центра. Представлял из себя, собственно, учебный компьютер, а также комплект учебной вычислительной техники (КУВТ). КУВТ состоит из рабочего места преподавателя (АРМП) и 12--16 рабочих мест учащихся (АРМУ).

[3] Эти уроки теперь входят в стандартную поставку KTouch. В дистрибутиве Debian (Etch) файл russian.ktouch.xml содержит имя автора: Chemisov Konstantin.

LINUX FORMAT
Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5500 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Родион Водейко, Елена Толстякова, Иван Мищенко
Переводчики
Илья Аввакумов, Александр Бикмеев, Светлана Кривошеина, Александр Казанцев, Алексей Опарин, Валентин Развозжаев, Александр Черных, Юлия Шабунио
Редактор диска
Александр Кузьменков
Допечатная подготовка
Мария Пучкова, Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

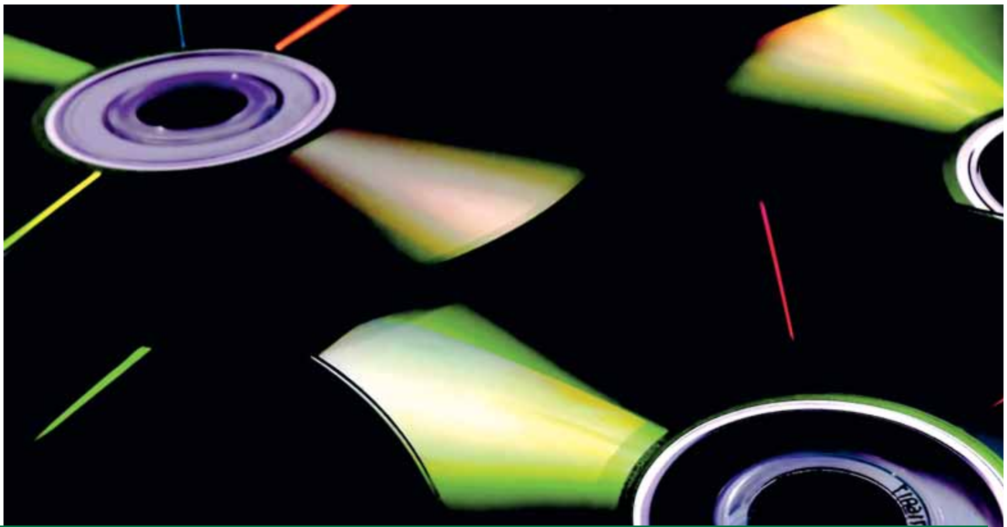
Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____

Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Редактор обзоров
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Литературный редактор
Ребекка Смолли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Ассистент по выпуску
Мэтт Нейлон (Matt Nailon)

Подготовка материалов
Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Д-р Крис Браун (Dr. Chris Brown), Энди Ченнел (Andy Channelle), Дэн Фрост (Dan Frost), Крис Хауэллс (Chris Howells), Евгений Балдин, Александр Бабаев, Андрей Боровский, Валерия Комиссарова, Галина Пожарина, Александр Поносов, Петр Семилетов, Алексей Федорчук, Александр Чернышов, Илья Шпаньков.
Художественные ассистенты: Сет Синх (Seth Singh), Роб Хейл (Rob Hale)
Фотографии: Дэвид Бланкенхорн (David Blankenhorn)
Иллюстрации: Нейл Бартлет (Neil Bartlett), Пол Блехфорд (Paul Blachford), Крис Винн (Chris Winn)

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция): ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Представительство в Москве:
пр.Мира, 161, телефон +7(495) 799-18-63, +7(495)136-88-45
Email: info@linuxformat.ru, Web: www.linuxformat.ru



в сентябре

Переустанови Linux!

Система тормозит и сыплется? Узнайте, как сохранить нужные данные и установить ее с нуля!

Верстаем в Linux

Да, Linux Format пока верстают на Макинтошах. Но, может у нас получится сверстать в Scribus хотя бы одну статью?

Тест для новичка

Три наугад выбранных человека с улицы — смогут ли они использовать Linux?



Ойвинд Колас

Автор GEGL проливает свет на свое детище.

Содержание последующих выпусков может меняться без уведомления